Commissioning of the ALICE PHOS Detector and Integration into the ALICE High Level Trigger

Per Thomas Hille

Thesis Presented for the Degree Philosophiae Doctor

Department of Physics
University of Oslo, Norway

January 2009

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
Nr. 857

ISSN 1501-7710

Cover: Inger Sandved Anfinsen.
Printed in Norway: AiT e-dit AS, Oslo, 2009.

Produced in co-operation with Unipub AS.
The thesis is produced by Unipub AS merely in connection with the
thesis defence. Kindly direct all inquiries regarding the thesis to the copyright
holder or the unit which grants the doctorate.

Unipub AS is owned by
The University Foundation for Student Life (SiO)

Preface

This thesis concludes my nearly six years of working with the ALICE PHoton Spectrometer (PHOS) starting with the testing of handful PHOS crystals in 2003 using hand soldered electronics and ending with the successful installation and commissioning of the first PHOS module in ALICE in September 2008.

I started in PHOS as a Master student in April 2003 obtaining my Master in 2005. I continued in PHOS, after a 1/2 year break, as a PhD student.

My PhD thesis has been carried out within the Norwegian part of the ALICE project in the Nuclear Physics Group at the University of Oslo, founded by the Norwegian research council. The work with the thesis has required permanent presence at CERN. The thesis was written in the period August 2008 to January 2009 at CERN.

The main focus of this thesis is the commissioning of PHOS. In order to test and commission a detector one has to be able to control and monitor it. These tasks are performed by the so called Detector Control System (DCS). When I started my PhD we had an upcoming electron-beam test but we had no DCS suitable for any large scale operation of PHOS. My first takes was therefore to implement the DCS for the PHOS front-end electronics.

During commissioning one wants to monitor the detector response and the data quality. A lot of time can be saved if this feedback can be provided on-line. Since PHOS is a detector that participate in the HLT it is natural that the HLT also provides the monitoring since the HLT has access to the full reconstructed event. Using the HLT for monitoring also serve another important purpose; The commissioning of the HLT software itself as the monitoring only works if the on-line event reconstruction works.

Not all commissioning data processing task are best suited for on-line processing in the HLT. One might want to scrutinize the data in ways that would require too much CPU time. Or one might want to scrutinize the same data several times using different tools or different configurations of the same tools, make figures etc. A dedicated PHOS (off-line) debugging library was therefore developed (see Section 9.2.1), complementing the features of the HLT monitoring.

Organization of the Thesis & my Contributions

Chapter 4 After the introduction in Chapter 1 - 3 the main part of my thesis starts with the description of the PHOS detector. Section 4.2 to 4.4 describes the main physics objectives of PHOS, the principle of operation and factors affecting the energy, time-of-flight and the position resolution. All important factors that determine the physical layout of PHOS which follows in Section 4.5.

Section 4.6 describes the PHOS front-end electronics with an emphasis on features relevant for the next Chapter describing the DCS system which in turn is relevant for Chapter 7 about calibration since the calibration of is PHOS partly performed by the configuration of the electronics (the APD bias control).

In addition I have chosen to put some emphasis on the frequency domain response of the PHOS shaper which is important background material for the understanding of the noise problems discussed in Chapter 9.

The last section about readout strategy is also relevant for PHOS DCS, which will configure the electronics according to this strategy, and for the HLT since the data rates depends, among other things, upon the configuration of the electronics.

Chapter 5 This chapter describes the DCS for front-end electronics as it is currently implemented for PHOS and how it is foreseen to be integrated in the final ALICE DSC system. All PHOS specific DCS software described in this chapter was developed by me except the *tru-scan* tool which was made by Øystein Djuvsland (see Section 5.6) using my DCS interface library to interface the electronics. In addition I should mention that *sc-testbench* tool is a re-implementation of a similar tool made by Jo-Inge Buskenes.

The user guide to the PHOS DCS software is included in Appendix J for completeness. This manual is the genuine final version of the user guide as it was/is used in the LAB. In order to keep it genuine I have not made any modifications except correcting typing mistakes.

Chapter 6 This chapter describes the process of identifying photons, all the way from the extraction of energy and time-of-flight from the readout channels, the subsequent clusterization and deconvolution of showers induced by different particles, and finally identification of photons using shower topology and time-of-flight.

I chose to dedicate a full chapter to the *Extraction of PHOS detector signal parameters needed for physics analysis*. Firstly, because this is obviously a very important subject since all subsequent analysis of PHOS data depends on it. Secondly, because I spent a lot of time and effort during my PhD investigating the first step in the process; The extraction of amplitude and peak position (see Section 6.2.1).

Furthermore most of the material found in Appendix G was originally intended to be included in the main part (Chapter 5 - 10) of this thesis. The idea was to go systematically

through the *desirable properties of estimators*, then compare the Least Mean Square and the Peak-Finder with these properties. However I found, as the work with my thesis developed, that including this material in the main part would make the thesis look somewhat unbalanced, focusing too much on *textbook* material and that the focusing on this material could shift focus away from the *non-textbook* material presented in Chapter 5, 6, 8, 7 and 9.

In addition I would like to add that the derivation of the optimum Peak-Finder vectors presented in Section G.1.5.1 was done by me, using common techniques from from statistical digital signal processing, However since only the simplified version of the Peak-Finder algorithm was used so far (see Section 6.2.1.3) this derivation was moved to Appendix G

Chapter 7 A proper calibration is of course very important for the physics performance of PHOS. Chapter 7 gives a brief overview of probes that can be used to calibrate PHOS. In the subsequent sections I give a brief overview of calibration methods and issues with an emphasis on material relevant for Chapter 9.

I have participated in various ways during electron beam-tests of a 256 ch prototype and the first PHOS module as well as the attempt to calibrate the first PHOS module using cosmic muons. In addition to my contribution to the DCS system, used to calibrate APD bias control, I have participated in the setting up of network infrastructure and the trigger system. I have taken a substantial number of shifts.

The importance of the subject combined with the fact that I have made several important contributions made it natural for me to devote a full chapter to this subject.

Chapter 8 Describes the PHOS HLT system the way it was implemented and used so far in ALICE and in the PHOS lab. I have also outlined a possible scenario on how the HLT for PHOS can be run in the final system.

A few snapshots from the on-line monitoring in ALICE are included in order to demonstrate the usefulness of the monitoring.

The HLT for PHOS has been my main responsibility during my PhD period. All PHOS specific HLT software was written by Øystein Djuvsland and me. Furthermore, the PHOS HLT on-line monitoring (see Section 8.5) was written by me.

Furthermore the benchmarking of the HLT software revealed that the bottleneck of the raw data processing was the decoding of the data. I therefore wrote a new decoder (see Appendix D) which is now the standard one for off-line and on-line processing.

Chapter 9 I have put the chapter about the commissioning of the first two PHOS modules last, although I regard this chapter as the most important part of my thesis. The reason is that material from all the preceding chapters is relevant.

The content of this Chapter reflects the fact there were a LOT of problems. I go through step by step most of the problems encountered, how some of them were solved,

and some of the problems led to modifications in parts of the design of PHOS.

I have actively participated in most aspects of the commissioning of the the first two PHOS modules. This involves, among other thing, the physical installation of infrastructure at ALICE. On the software side I have made a collection of debugging tools used during the commissioning of the second PHOS module. These tools are described in Section 9.2.1 and are all written by me, using ROOT to display the plots.

Acknowledgements

There are several people who deserve acknowledgement for their contribution to the work compiled in this thesis. My sincerest thanks goe to: my supervisors Trine Tveter, Bernhard Skaali and Dieter Röhrich, for excellent guidance and support during all stages of my thesis.

Thanks to Øystein Djuvsland, my successor in PHOS, for numerous social activities, and for sharing my passion for physics, programming and for getting stuff to work. Our collaboration in PHOS has been remarkably productive. Without him the successful implementation of the PHOS HLT software would not have been possible.

Thanks also To Jochen Thaeder, my flat mate, for sharing movie nights and other social activities together.

I also wish to thank Johan Alme for numerous social activities and for a very productive collaboration during the commissioning of the PHOS front-end electronics.

Furthermore I would like to thank several of my colleagues (in alphabetical order) who have contributed in various ways, both with technical issues and social activities, to make my PhD period at CERN such a rewarding experience: Kenneth Aamodt, Sebastian Bablok, Lamia Benhabib, Øystein Haaland, Haavard Helstrup, Yuri Kharlov, Stefan Kirsch, Dag Toppe Larsen, Mattias Richter, Boris Polischchuk, Therese Sjursen, Timm Steinbeck and Alexander Vinogradov. The list is probably not complete and I apologize if I out of forgetfulness or ignorance has omitted someone.

Finally, I am deeply in debt to Erika, for encouragement and for having absolute confidence in me.

Oslo, Norway, January 2009
Per Thomas Hille

Contents

List of Figures

[1] The modified version taken from [7] is used

List of Tables

Abbreviations and definitions

ACR	=	Alice Control Room
ADC	=	Analog to Digital Converter
AIF	=	ALTRO Interface, or ALTRO Interface module (Firmware module in the RCU main FPGA)
ALICE	=	A Large Ion Collider Experiment
ALTRO	=	ALICE TPC ReadOut chip
APD	=	Avalanche Photo Diode
CERN	=	European Organization for Nuclear Research
CERN GPN	=	CERN General Purpose Network
CN	=	Computing Node (In the High Level Trigger)
CPV	=	Charged Particle Veto detector
CSP	=	Charge Sensitive Preamplifier
CTP	=	Central Trigger Processor
DAC	=	Digital to Analog Converter
DATE	=	ALICE Data Acquisition and Test Environment
DCS	=	Detector Control System
DCSC	=	Detector Control System interface Card
DDL	=	Detector Data Link
d.f	=	Degrees of Freedom
DIU	=	Data Interface Unit
D-RORC	=	Detector ReadOut Receiever Card
ESD	=	Event Summary Data
FED	=	Front End Device
FEP	=	Front End Processor (in the High Level Trigger)
FMD	=	Forward Multiplicity Detector
FEE	=	Front End Electronic
FEEC	=	Front End Electronic Card
FIR	=	Finite Impulse Response
FSM	=	Finite State Machine
GDC	=	Global Data Concentrator
GPN	=	General Purpose Network

GTL = Gunning Transceiver Logic (bus)
HG = High Gain
HMPID = High Momentum Particle Identification Detector
HOMER = HLT On-line Monitoring Environment including ROOT
HTML = Hyper Text Markup Language
i.i.e = independent and identical errors
ITS = Inner Tracking System
JPEG = Joint Photographic Experts Group
L0 = Level 0 (trigger)
L1 = Level 1 (trigger)
LAB = Laboratory
LDC = Local Data Concentrator
LED = Light Emitting Diode
LEP = Large Electron Positron collider
PF = Peak Finder (algorithm)
PS = Proton Synchrotron
PS = Power Supply
LG = Low Gain
LHC = Large Hadron Collider
LM = Levenberg Marquardt (nonlinear fit)
LSI = Linear and Shift Invariant (FIR)
LTU = Local Trigger Unit
MAC = An Apple Macintosh computer
mvlb = minimum variance lower bound
MPEG = Moving Picture Exprts Group
OCDB = Offline Condition Database
PABC = PHOS APD Bias Control
PDS = Permanent Data Storage
PHOS = PHOton Spectrometer
PID = Particle Identification
PMD = Photon Multiplicity Detector
pRORC = PCI Readout Receiver Card
PS = Proton Synchrotron
QCD = Quantum Chromo Dynamics
QED = Quantum Electro Dynamics
QGP = Quark Gluon Plasma
RCU = Read out Control Unit
RIO = Region Of Interest
SC = Slow Control
SCIF = Slow Control Interface or Slow Control Module
SIU = Source Interface Unit

SPS	=	Super Proton Synchrotron
TDA	=	Time to Amplitude Converter
TDC	=	Time to Digital Converter
TDS	=	Transient Data Storage
TOF	=	Time Of Flight
TRD	=	Transition Radiation Detector
TPC	=	Time Projection Chamber
TRU	=	Trigger Router Unit
VO	=	Virtual Organization
ZDC	=	Zero Degree Calorimeter

Part I

Introductory Overview

Introduction

The *standard model* of particle physics seems to form a microscopic basis for all known physical phenomena except gravity. The aim of ultra-relativistic heavy ion collisions is to study how collective phenomena and macroscopic properties of strongly interacting matter emerge from the microscopic laws of the strong force [15].

Lattice calculation of Quantum Chromo Dynamics (QCD) predicts that at a critical temperature of 170 MeV nuclear matter undergoes a phase transition to a deconfined state of quarks and gluons denoted the Quark Gluon Plasma (QGP). In heavy ion collisions at the LHC one expects to reach and exceed the energy density necessary to create the QGP. The main objective of the ALICE experiment is to study the physics of the QGP state thus obtaining a better understanding of the standard model at low energy and the evolution and large scale structure of our universe.

A direct link between the predictions of the standard model and observables in heavy ion collisions exists only for a limited number of observables. Photons, since they do not interact strongly, are interesting probes for exploring the QGP. The mean free path of the photons in the QGP is long compared to the size of the system so that the photons carries important information about all phases of the evolution of the QGP. The role of the PHOton Spectrometer (PHOS) in the ALICE experiment is to detect and identify photons and measure with high resolution their 4-momentum with the prospect of performing the ALICE physics program of exploiting direct photons and light neutral mesons π^0 and η mesons mainly) as probes [15]. In addition jet quenching will be studied via high p_T π^0 's and together with EMCAL and TPC via γ-jet and γ-hadron correlations using EMCAL and PHOS as trigger.

An additional challenge is to select rare events in the presence of a huge background of ordinary events. It was decided by the ALICE collaboration that the objectives of the physics program could be achieved at an acceptable cost with a storage bandwidth of 1.25 GB/s whereas the data selected by the trigger system amounts to more than 20 GByte/s. Further compression of the data will be performed by the High Level Trigger (HLT). By sophisticated on-line analysis the available bandwith will be set aside for the most relevant physics events.

Chapter 1

Ultra-relativistic Heavy-Ion Collisions

The main objective of high energy heavy-ion physics is to apply and extend the standard model to complex and evolving systems. Ultra-relativistic heavy-ion physics addresses this question in the sector of the strong interaction by studying nuclear matter at extreme density and temperature [15]. At the LHC the energy will be high enough to produce the QGP thereby providing the possibility to study the QGP.

1.1 Hot and Dense Nuclear Matter

According to our current understanding of *Big Bang* cosmology our universe began as an infinitely hot and dense fireball. This fireball then expanded explosively and its temperature and density has continued to decrease until the present day [16]. The early universe is believed to have undergone three major phase transitions. The only one of these phase transitions believed to be reachable by any laboratory experiment on earth is the transition between a state where the quarks behave as free particles into a state where the quarks are bound into hadrons. By understanding the physics of the QGP one can obtain a better understanding of the origin of mass in our universe. In order to study the QGP one has first of all to prove its existence and that it is actually created during for instance Pb+Pb collisions at LHC. Once the existence is proved one can investigate the properties of the QGP.

Figure 1.1 gives a schematic overview of the nuclear phase diagram. The *liquid* phase is realized in atomic nuclei at zero temperature. At low densities the nucleons behave as a gas.

As the temperature increases the nucleons are exited into hadronic resonances which subsequently decays into pions and nucleons. This *hadronic matter* is indicated by the white area in the figure. At even higher temperatures the hadron density increases mainly

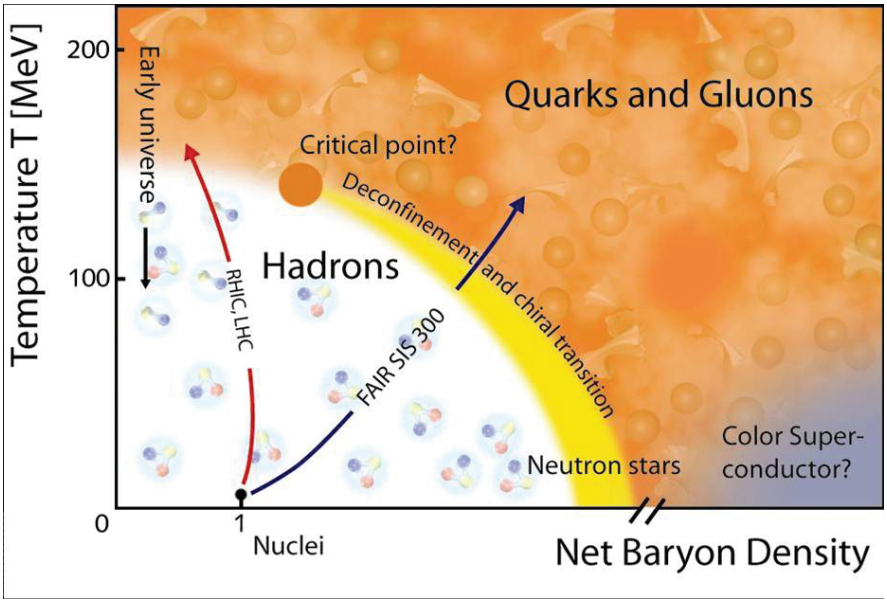


Figure 1.1: The phase diagram of strongly interacting nuclear matter [1].

due to pion production. At a temperature of around 200MeV, and at baryon densities on the order of the density of ordinary nuclear matter, the hadrons interact so frequently and violently that one can no longer assign individual quarks or gluons to a particular hadron. The hadrons disintegrate into quarks and gluons and the system enter the QGP phase.

The QGP phase in the limit of low baryon density and high temperature is assumed to have been the state of the early universe up to 10^{-6} s after the big bang. Understanding the phase transition/crossover in this region is therefore crucial in order to understand the evolution of the early universe.

As seen from Figure 1.1 the QGP phase transition can also occur at low temperature and high baryon density if the density is increased at zero temperature to more that $\sim$10 times that of ordinary nuclear matter. The phase transition at zero degrees can be understood as follows: At normal densities each nucleon occupies a volume of around 6 fm^3 whereas the size of the nucleon itself is approximately one tenth of this. Therefore, if the baryon density is increased by a factor $\sim$10 the wave functions of the quarks overlap enough that one can no longer assign a quark to an individual nucleon and the quarks can be said to move *freely* within the entire nuclear volume [16]. QGP in the limit of low temperature and high baryon density is believed to be state of matter for instance in the center of a neutron stars.

The Critical Point The phase transition to the QGP is believed to be of first order only up to a certain point, the *critical point*, on the nuclear phase diagram. Referring to Figure 1.1 phase transitions occurring to the right of the critical point have a discontinuity in the energy density on the boundary between the hadron and the QGP phase and is consequently of first order. Phase transitions occurring to the left of the critical point have a continuous energy density and have a transition region rather than a transition boundary. The exact location of the critical point and whether a QGP phase transition will occur to the left or right of the critical points at LHC energies is an unresolved question.

1.2 The Dynamics of Heavy Ion Collisions

Heavy ion collisions are generally divided in two energy domains: Lower energies where the stopping power is sufficient to stop the colliding nuclear matter, and higher energies, where the colliding baryons initially penetrate each other. The latter scenario is most likely to be the case at LHC energies, and is often described with the hydrodynamical model of Bjorken [17].

Due to the Lorenz contraction in the longitudinal direction the two colliding nucleons is seen, in the reference frame of the experiment, as two two-dimensional sheets of coloured partons, mainly gluons. The nuclei become almost completely transparent to each other,

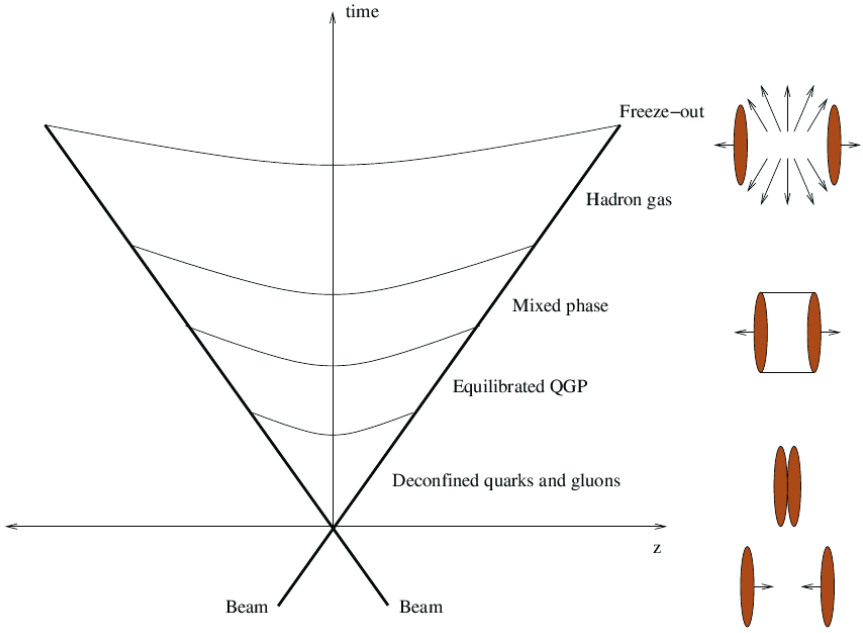


Figure 1.2: Space time diagram for heavy ion collisions [2].

and the valence quarks maintain their initial rapidities. At the inter-penetration of the two nuclei $q\bar{q}$ pairs materializes from the gluon field creating a non-equilibratet and deconfined system of quarks and gluons. This system is called the *pre-equilibrium* phase.

If the matter created in the reaction zone re-scatters efficiently then the system will reach a state of thermal equilibrium after a formation time in the range 0.5 - 2 $^{\text{fm}}/_{\text{c}}$. From that time on the system can be treated as an ideal relativistic fluid as indicated in Figure 1.2. collisions. Referring to the figure the projectile nucleus comes from z = -∞ and the target nucleus from z = -∞. The two nuclei meets at z = 0, t = 0. Immediately after the collision, if the energy is high enough, the nuclear matter is believed to enter the QGP state. As the system expands and cools the temperature is no longer high enough for the QGP to exists and one gets a phase transition from the QGP to a state where the quarks are confined in hadrons denoted *hadronization*. If the transition from the QGP to a hadronic gas is of second order the systems will spends some time in a mixed, or *cross-over*, state where the QGP coexists with the hadron gas. Finally, the hadronic *freeze-out* occurs when the system has expanded enough for the mean free path of the hadrons to become large compared to the size of the system. From the *freeze-out* on all the constituent of the fluid is considered to be independent and final interactions are

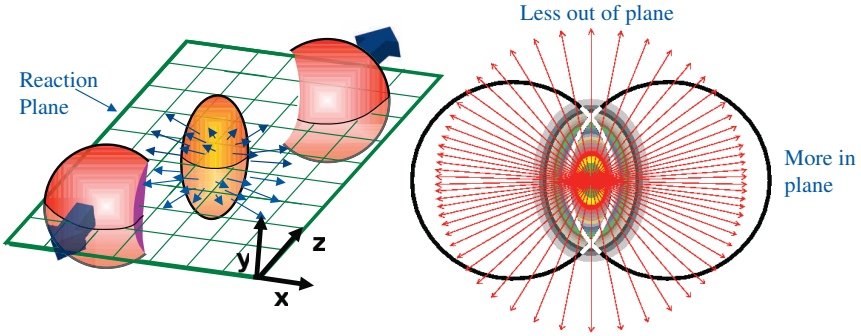


Figure 1.3: Elliptic flow: the denser production of particles in plane with respect to out of plane is due to a higher pressure gradient in plane.

neglected.

Only signals surviving until the freeze-out can be measured. The general procedure to describe what happens during a heavy ion collision is to measure as many outgoing particles as possible and describe their behaviour in terms of kinematic variables such as rapidity and transverse momentum. One way to characterize the results is in terms of the cross section of various produced particle species. Results from p+p collisions are typically used as reference for the interpretation of A + A results where A denote a target and a projectile with atomic number A for instance Pb + Pb.

1.3 Signatures of the Quark Gluon Plasma

It is believed that there is no single unique signal that can be used to prove the existence of the QGP, but rather one has to accumulate observations on several different signatures [18]. Some of these signatures are listed below.

Elliptic Flow In non-central heavy-ion collisions, the initial overlap zone between the colliding nuclei is spatially deformed. Strong rescattering during the early, partonic phase will lead to pressure gradients and a transformation of the initial-state anisotropy in coordinate space into a final-state anisotropy in momentum space (see Figure 1.3). The momentum anisotropy can be described by the equation:

$$E\frac{d^3N}{d^3p} = \frac{1}{2\pi}\frac{d^2N}{p_{\mathrm{T}}dp_{\mathrm{T}}dy}\left\{1 + 2\Sigma_{n=1}^{\infty} v_n(p_{\mathrm{T}}, y)\cos[n(\phi - \Psi_R)]\right\} \tag{1.1}$$

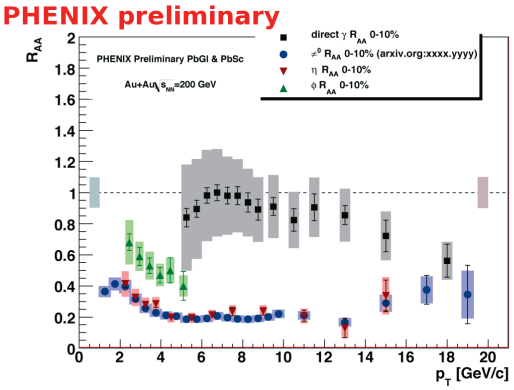


Figure 1.4: Preliminary results from RHIC. Light mesons are strongly suppressed compared to direct photons as for p+p compared to Au + Au at $\sqrt{s} = 200$ GeV [3].

where ϕ and Ψ_R are the azimuthal angles of the particle and the reaction plane, respectively. Recent results at RHIC have revealed that the elliptic flow coefficient v_2 reaches values close to those expected for a perfect fluid and that v_2 behaves differently for mesons and baryons, scaling with the valence quark content. This indicates a strongly coupled state of matter with partonic degrees of freedom. A review of recent developments in the field of anisotropic flow is given in [19].

High p_T Suppression Coloured partons (quarks and gluons) with high p_T, moving through a medium with free colour charges, will undergo multiple interactions and lose energy, both through medium-induced gluon radiation and through collisions. Using data from proton-proton collisions as a reference, medium effects can be quantified through a so-called nuclear modification factor, defined as

$$R_{AA} = \frac{d^2N_{AA}/dydp_{\mathrm{T}}}{\langle N_{binary}\rangle d^2N_{pp}/dydp_{\mathrm{T}}} \tag{1.2}$$

where $\langle N_{binary}\rangle$ is the average number of binary nucleon-nucleon collisions in the A+A reaction. In the absence of medium effects, A+A collisions are expected to be merely a superposition of elementary nucleon-nucleon-collisions, and the nuclear modification factor would be unity. However, results from RHIC show strong suppression of high-p_T hadrons in A+A collisions compared to p+p collisions ($R_{AA} < 1$), refer to Figure 1.4, Photons, which do not interact strongly, have a long mean free path in the QGP and are therefore not suppressed (except at very high p_T, possibly due to isospin effects). Nuclear modification factors were determined also for d+Au collisions, where cold nuclear matter, and possible initial-state effects, like shadowing / gluon saturation, should be present. At

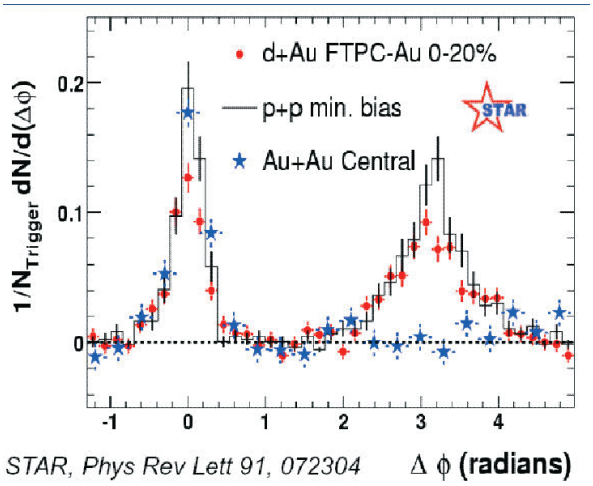


Figure 1.5: Jet quenching measured in p+p, d+Au and Pb+Pb at STAR [4].

midrapidity the measured R_{dA} showed enhancement instead of suppression, confirming that the quenching of high-p_T hadron spectra seen in Au+Au is indeed a final-state effect [3].

Jet Quenching (di-hadron correlations) This is essentially the same phenomenon as high p_T suppression. High-p_T hadrons are mostly produced in hard QCD scattering during the early phase of the reaction, resulting in two energetic partons moving nearly back-to-back in the azimuthal direction, with equal but oppositely directed transverse momenta (dijet events). They eventually fragment into hadrons. Jet quenching can be studied through azimuthal correlations, where a high-p_T hadron (often the leading hadron in a jet, or in an event) is chosen as the trigger particle. The differential yield of (usually softer) associated particles is studied as a function of their difference in azimuthal angle, $\Delta\phi$, relative to the trigger particle. The high-p_T hadrons emitted close to $\Delta\phi = 0$ are denoted as the near-side jet, while the away-side jet is centered at $\Delta\phi = \pi$. If such a parton pair is created within the QGP, one of the partons will usually have a shorter path length through the medium and undergo less energy loss, typically ending up as the near-side jet with higher leading-particle p_T. The other parton will have to travel a longer path through the plasma and suffer more interactions, leading to a shift in the parton transverse momentum and broadening of the resulting away-side jet. Correlations between high p_T hadrons at RHIC [4], show that the away-side jet is strongly suppressed in central Au-Au collisions at $\sqrt{s_{NN}} = 100$ GeV, but not in p-p or d-Au (see Figure 1.5). This may be taken as an indication that a medium with free colour charges is formed in

the final state of central Au-Au collisions.

Strangeness Enhancement In a QGP, the effective strange quark masses will be lower than in a hadronic medium, due to the restoration of chiral symmetry, enhancing the thermal production of $s\bar{s}$ pairs. Also in partonic matter, formation of strange and in particular multi-strange hadrons can take place by simple quark coalescence, instead of the multistep hadronic collisions with high energy thresholds that are required in a hadron gas. A strong enhancement of the yield of strange particles per participant, in particular for multi-strange baryons, has been observed at SPS in central Pb-Pb collisions relative to p-Be collisions [20]. However, at least part of the enhancement can probably be explained by canonical suppression of strange particle production in small systems.

J/Ψ Suppression/Enhancement Charm-anticharm quark pairs are produced in hard initial partonic collisions and may form one of several charmonium states (J/Ψ,Ψ',χ_c). Characteristic of the deconfined state of matter is the dynamic screening of the long-range confining QCD potential. Colour screening is predicted to reduce the attraction between heavy quark-antiquark pairs, and leads to a suppression of J/Ψ yields. Measurements in Pb-Pb reactions at $\sqrt{s_{NN}} = 17.3$ GeV by the NA50 experiment [21] show a suppression of heavy charmonia production relative to "normal" nuclear absorption, dissociation of $c\bar{c}$ pairs by interactions with nucleons. The suppression is found to be similar at RHIC and SPS, despite a denser medium at the higher energy. At RHIC the J/psi suppression even falls at forward rapidity, although the medium should be more dilute there than at midrapidity [22]. Hence the status of J/psi suppression as a QGP signal is quite ambiguous. At LHC, a possible QGP signal could be the recombination of originally uncorrelated heavy quark-antiquark pairs which are expected to be copiously produced at these energies, and an associated enhancement of charmonium states.

1.3.1 Electromagnetic Signatures

Several mechanisms of photon production are at work in ultrarelativistic heavy-ion collisions [23], [24] [25]:

During the initial stage of interpenetration of the partonic clouds of the colliding nuclei, prompt photons are produced by the hard scattering of quarks and gluons. The leading order processes are Compton scattering ($q + g \rightarrow q + \gamma$) and annihilation ($q + \bar{q} \rightarrow g + \gamma$). The corresponding p_{T} spectrum has a power-law shape, and photons up to several hundred GeV can be produced. Another source of prompt photons is the fragmentation of high-p_T quarks. Very energetic partons later hadronizing as π^0 (decaying electromagnetically) are also formed at this stage.

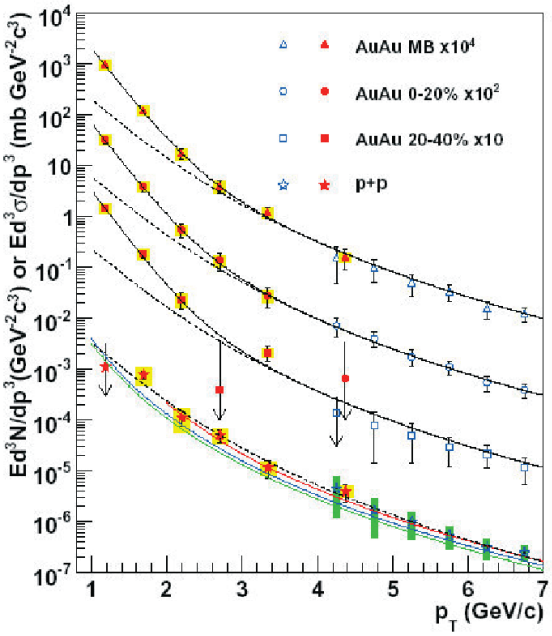


Figure 1.6: Direct photons measured at PHENIX[5].

A thermally equilibrated bubble of QGP is formed, with an initial temperature of up to 1 GeV. Thermal photons are produced through various processes, including Compton scattering, quark-antiquark annihilation and bremsstrahlung. The spectrum is exponentially suppressed but should extend up to several GeV. The detailed shape of the energy spectrum depends on the hydrodynamic evolution of the fireball.

Recent calculations suggest that hard partons passing through the plasma may induce significant production of high energy photons, through annihilation and Compton scattering on thermal partons, plus bremsstrahlung during interactions of quark jets with the plasma. Photons from this source will typically give a harder spectrum component than the thermal radiation from the QGP, and may make a large contribution at p_T ~4 - 8 GeV/c. This component has still not been verified experimentally, but is predicted to be more important at LHC energies.

After hadronization, photons can be emitted in the scattering of mesons or in resonance decays. The energy typically ranges from a few hundred MeV to several GeV.

The shape of the photon p_{T} spectra both in the plasma and hadron gas phase may be modified by expansion. This thermal photon spectrum from the hadronic phase is most likely outshone by the QGP component.

Finally after freezeout, a huge photon background is produced through decay of neutral mesons, mostly π^0 and η.

Virtual photons, emerging as dileptons with opposite charges, are produced through similar processes as real photons. These and a number of additional dilepton sources are discussed in refs.[26], [27]. The detection of leptonic daughters of virtual photons provides a useful complement to the measurements of real photons.

The PHENIX experiment at RHIC measured both real and virtual photons (the latter as e^+e^- pairs) from Au-Au collisions with $\sqrt{s_{NN}} = 200$ GeV and p-p collisions at $\sqrt{s} = 200$ GeV. Combining the two measurements after subtraction of mesonic decay contributions, a significant excess of photons was found for both systems (see Figure 1.6). The photon spectra from the p+p collisions have a power-law shape consistent with pQCD predictions. The spectra from Au+Au collisions were well described by a two-component ansatz, with the pQCD power-law spectrum from the p+p system scaled to the correct number of binary nucleon-nucleon collisions, plus an exponential component extending up to $p_T \sim 3$ GeV/c, consistent with expectations for the QGP thermal spectrum.

Chapter 2

The ALICE experiment at LHC

2.1 Introduction

2.1.1 The LHC Experiments

The central question addressed by the LHC experimental program is the origin of mass. Figure 2.1 shows the CERN accelerator complex and the four main experiments; ATLAS, CMS, ALICE and LHCb.

ATLAS and CMS will search for the Higgs particle, which generates the mass of the electroweak gauge boson and the bare mass of elementary fermions through spontaneous breaking of the electroweak gauge symmetry.

They will also search for supersymmetric particles which are manifestations of a broken intrinsic symmetry between fermions and bosons. LHCb will study CP-symmetry violating processes, focusing on precision measurements with heavy b quarks.

ALICE will study the role of chiral symmetry in the generation of mass in composite particles using heavy-ion collisions to attain energy densities high enough to create a deconfined state of quarks and gluons, the QGP, over large volumes and timescales. ALICE will study the properties in general as well as the time evolution of the hot partonic medium.

2.1.2 The LHC Running Strategy

The LHC will start each year running for several months with proton beam followed by several weeks of Pb+Pb collisions. The total running time, the luminosity ($\mathscr{L}$) and the maximum $\sqrt{s}$ is given in Table 2.1 Since ALICE is a dedicated heavy-ion experiment it is not suited to cope with the nominal LHC luminosity for p+p. The beam will therefore be de-focused in the ALICE interaction region to obtain a luminosity of $\mathscr{L} = 10^{30} cm^{-2} s^{-1}$.

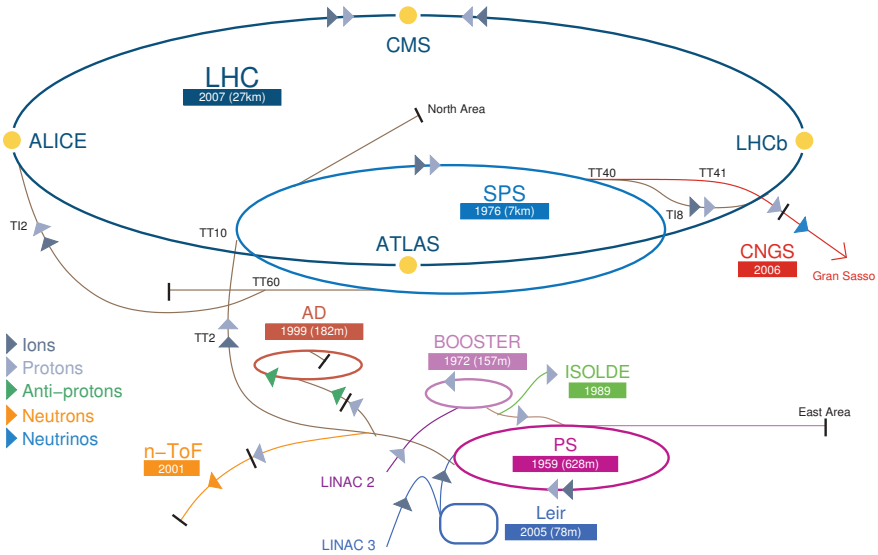


Figure 2.1: The CERN accelerator complex [6], ©CERN Copyright [1].

System	Total running time /year	$\mathscr{L}$	$\sqrt{s}_{max}$
p+p	$\sim 10^7 s$	$10^{34} cm^{-2} s^{-1}$	14 TeV
Pb+Pb like (Section 2.5)	$\sim 10^6 s$	$10^{27} cm^{-2} s^{-1}$	5.5 TeV

Table 2.1: Running time, luminosity and maximum centre of mass energy for p+p and Pb+Pb collisions.

2.1.3 The ALICE Experiment

ALICE is a general-purpose heavy-ion experiment focusing on the quantitative study and the time evolution of the hot partonic matter. ALICE is designed to cope with the highest particle multiplicities anticipated for Pb+Pb collisions (dN_{ch}/dy up to 8000). In addition ALICE will study data taken with lower-mass ion beams, which are means of varying the system size and energy density, and proton beams (both pp and pA), which primarily provide reference data for the nucleus-nucleus collisions [28]. Among the most interesting signals from the early stage are collective anisotropic flow, and hard and electromagnetic tomographic probes like jets, direct photons, heavy flavored hadrons and quarkonia, and their interaction with the medium. A particular challenge is disentangling initial and final state effects.

2.2 Photon Physics in ALICE

Photons are interesting probes, being produced through all phases of the reaction by a variety of processes (see Section 1.3). Since they do not interact strongly they carry information about all stages of the evolution of the QGP.

There are three detectors dedicated to photon physics in ALICE; PMD, PHOS, and EMCAL.

PMD The PMD addresses physics issues related to event-by- event fluctuations, flow and formation of Disoriented Chiral Condensates (DCC) and provides estimates of transverse electromagnetic energy and the reaction plane on an event-by-event basis [15]. Most of the photons detected by the PMD are decay photons from neutral mesons, mainly π^0 .

EMCAL The EMCAL is a large acceptance Electromagnetic Calorimeter covering the acceptance $|\eta| < 0.7$, $0 < \phi < 2\pi/3$. EMCAL is a detector dedicated to the study The main objective of the EMCAL is to improve ALICE's capability for studying jet physics, allowing detection of neutral hadronic and electromagnetic jet components, in addition to the charged particles measured in the barrel detectors. It has excellent momentum resolution for charged particles in the range 100 MeV/c to 100 GeV/c and covers nearly the full range of fragmented momentum for the highest energy jest accessible in heavy ion collisions [29].

PHOS The main objective of PHOS is to measure direct photons and light mesons. In addition jet quenching will be studied via high p_T π^0 s and in combination with the TPC and EMCAL via γ-jet and γ-hadron correlations, with the hard trigger photon being detected by PHOS. A detailed description of PHOS is given in Chapter 4.

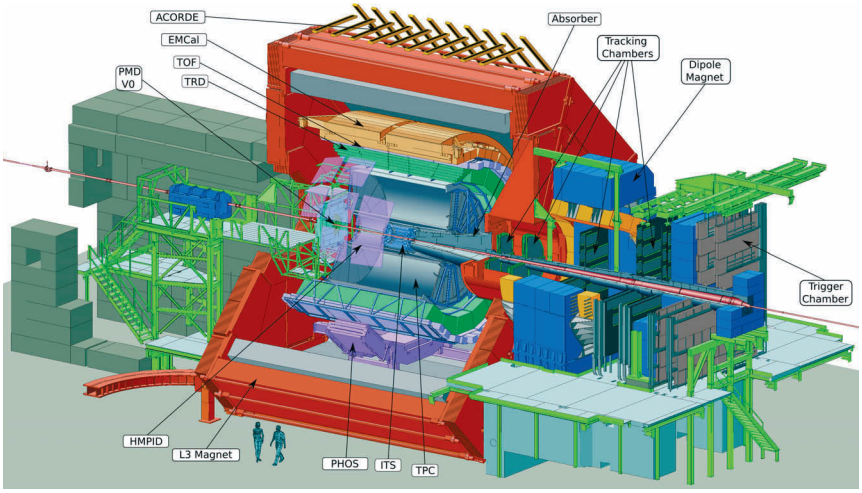


Figure 2.2: The ALICE detector.

2.3 Detector Layout

The ALICE detector is shown in Figure 2.2. The three photons detectors, PMD, EMCAL and PHOS were already described in Section 2.2. The other sub-detectors will be described briefly in this section. A short description of the detector layout and the physics objectives of the ALICE experiment is presented in this section. A detailed descriptions can be found in the ALICE Physics Performance Report [15] [28], and for the detector specific Technical Design Reports [30] [31] [32] [33] [34] [35] [36] [14] [37] [29]. [38]

Later several detectors have been modified based on recent discoveries at RHIC and new developments in detector physics, these modifications are described in [39] [40].

2.3.1 Central Barrel Detectors

The central barrel detectors consists of the two tracking detectors, the ITS and the TPC, and the two PID detectors TRD and TOF. They cover the rapidity range $|\eta| < \sim 0.9$ over the full azimuthal angle.

2.3.1.1 Tracking Detectors

Inner Tracking System (ITS) The ITS consists of three sub-detectors, starting from the centre and going outwards: the silicon pixel detector (SPD), the silicon drift detector (SDD) and silicon strip detector (SSD). Each sub-detector has two layers. The ITS is

optimized for effective track finding and high impact parameter and vertex resolution [35].

Time-Projection Chamber (TPC) The TPC is the main tracking detector of the ALICE central barrel. Together with the ITS it has to provide charged particle momentum measurements with good two track separation, particle identification, and vertex determination [30].

2.3.1.2 PID

Transition-Radiation Detector (TRD) The main goal of the TRD is to provide electron identification in the central barrel for momenta greater than 1 GeV/c, where the pion rejection capability through energy loss measurement in the TPC is no longer sufficient. The TRD will provide, along with data from ITS and TPC, sufficient electron identification to measure the production of light and heavy vector-meson resonances and the di-lepton continuum in Pb-Pb and pp collisions [31].

Time Of Flight (TOF) Detector TOF will together with the ITS and TPC detector provide PID of pions, kaons and protons in the intermediate momentum range from 0.2 to 2.5 GeV/c [32].

High-Momentum Particle Identification Detector (HMPID) HMPID is dedicated to inclusive measurement of identified hadrons for $P_T > 1$ GeV/c. HMPID will enhance the PID capability of the ALICE experiment by enabling identification of particles beyond the momentum interval attainable through energy loss (ITS and TPC) and time-of-flight measurement in TOF [37].

2.3.2 Forward Detectors and Event Characterization

Muon Spectrometer The complete spectrum of heavy quark vector mesons will be measured in the $\mu^+ \mu^-$ decay channel. The simultaneous measurement of all the quarkonia species with the same apparatus will allow a direct comparison of their production rates as a function of different parameters such as transverse momentum and collision centrality [34].

Zero-Degree Calorimeter (ZDC) The zero degree forward energy decreases with increasing centrality. The ZDC determines the geometry of the collision and the event plane from the directed flow (v_1) signal by measuring the energy carried by spectator neutrons in the forward direction [36].

Forward Multiplicity Detector (FMD) The main functionality of FMD is to provide charged-particle multiplicity information in the pseudo rapidity range -3.4 $< \eta <$ -1.7 and 1.7 $< \eta <$ 5.1. FMD will allow for the study of multiplicity fluctuations on an event by event basis and determination of the reaction plane which both is important for flow analysis [38].

V0 Detector The V0 detector provides a minimum bias trigger for central barrel detectors and two centrality triggers in Pb + Pb collisions. In addition V0 gives validation signal for the muon trigger to filter background in pp collisions [38].

T0 Detector T0 generates a t_0 signal for the TOF detector and a *wake up* signal for the TRD. T0 measures in addition the vertex position with a precision of +/- 1.5 cm and gives an estimate of the particle multiplicity to generate one out of three possible trigger signals: $T0_{min-bias}$, $T0_{semi-central}$, or $T0_{central}$ [38].

2.4 ALICE On-line Systems

Figure 2.3 shows a simplified overview of the ALICE on-line systems with an emphasis on PHOS. The on-line systems are divided into the four domains DAQ, DCS, Trigger and HLT. The synchronization between these on-line systems is done by the Experiment Control System (ECS).

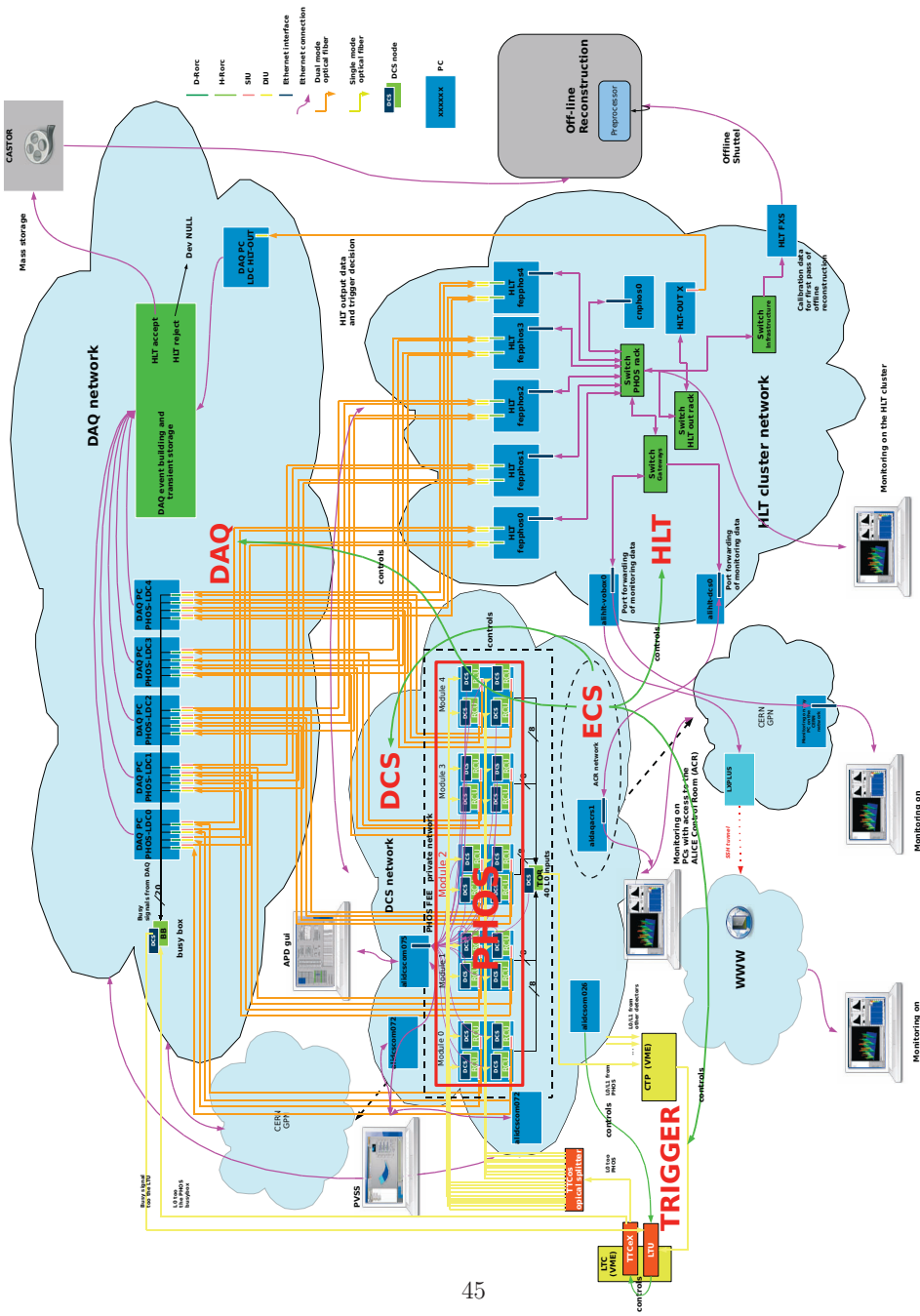
2.4.1 The Experiment Control System (ECS)

The control software follows the architecture of the on-line systems and is correspondingly divided into four main activity domains: HLT, trigger, DCS and DAQ. This pattern is repeated by all the sub-detectors[2]. The control software for the four activity domains are interfaced via Finite State Machines (FSM) with a well defined number of states and a set of actions (commands) associated with each state.

The ECS coordinates and synchronizes all the activity domains existing in ALICE and between these domains and the external world (e.g. LHC status, beam on/off etc..). From the point of view of the FSMs, the activity domains are unaware of each-other‘s existence and can communicate with each other only via ECS.

Figure 2.4 shows the FSM representation of the four activity domains. Each activity domain is broken up in several sub system in a tree like structure of parent child relationships. A child always has exactly one parent whereas a parent can have several children. The child communicates only with its parent. Communication with other children at the

[2]For the sub-detectors that do not participate in the HLT the activity domains are DCS, trigger and DAQ



45

Figure 2.3: Data flow and connectivity between the on-line systems at ALICE with an emphasis on PHOS

.

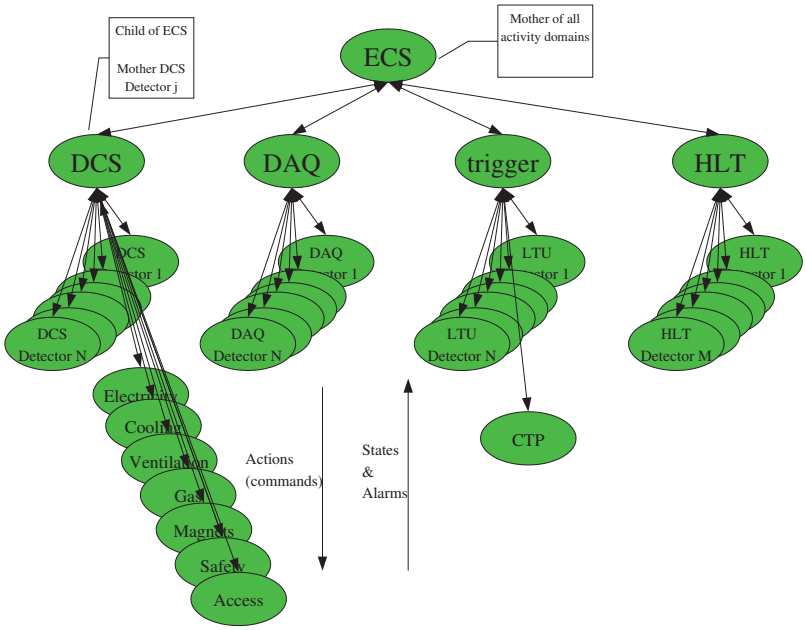


Figure 2.4: Top level FSM representation of the four activity domains DCS, DAQ, trigger and HLT. The synchronization between the activity domains is performed by ECS. The design follows a strict tree like hierarchical structure with a mother / child relationship between the nodes. As shown in the picture a mother can have several children whereas a child always has exactly one mother.

same level is forbidden. Commands are issued from the mother to the child but never from child to mother. The responsibility of the child is to execute the commands issued by the mother and report its state, which is only visible for the mother. A node can act both as a child and as a mother. For instance the DCS activity domain is a child of ECS while a mother for the detector specific DCS nodes. In addition any node can issue an alarm which is visible for any other node.

The FSM nodes below each activity domain can be separated in detector specific FSMs and experiment wide FSMs. As an example, in the trigger domain, there is a Local Trigger Unit (LTU) specific to each detector that correspondingly is modelled by detector specific LTU trigger nodes (FSMs) whereas the Central Trigger Processor (CTP) is common to the ALICE experiment as a whole. [8]

2.4.2 The Detector Control System (DCS)

The primary task of the DCS is to provide configuration, remote control, and monitoring of all experimental equipment. Unlike the other on-line systems the DCS is supposed to be operational, some times unattended, during all operational phases of the experiment, including shutdown periods.

In the rest of this thesis the terms *configuration*, *control* and *monitoring* shall have the meanings listed below.

Configuration Configuration of experimental equipments in order to make them ready for data taking. For example, during the configuration of the PHOS front-end electronics the ALTRO chips are configured for the number of samples to take on the arrival of a trigger, whether or not to use zero suppression etc. (see Section 4.7.1.1).

Monitoring The monitoring of parameters (typically temperature, voltages and currents) relevant for calibration and/or the safe operation of the detector and infrastructure. Using PHOS as an example, there are temperature sensors situated on the readout electronics which should be monitored to ensure that the temperature is low enough not to cause any damage to the electronics. In addition there are temperature sensors inside the cold volume containing the crystals (see Section 4.5). Fluctuations in these temperatures should be taken into account during event reconstruction.

Control The front-end electronics, for instance, is configured by issuing a command (typically from the supervisory layer) to the control layer (please refer to Figure 5.2) to configure the electronics. The actual configuration is then subsequently performed automatically by the control layer and the field layer.

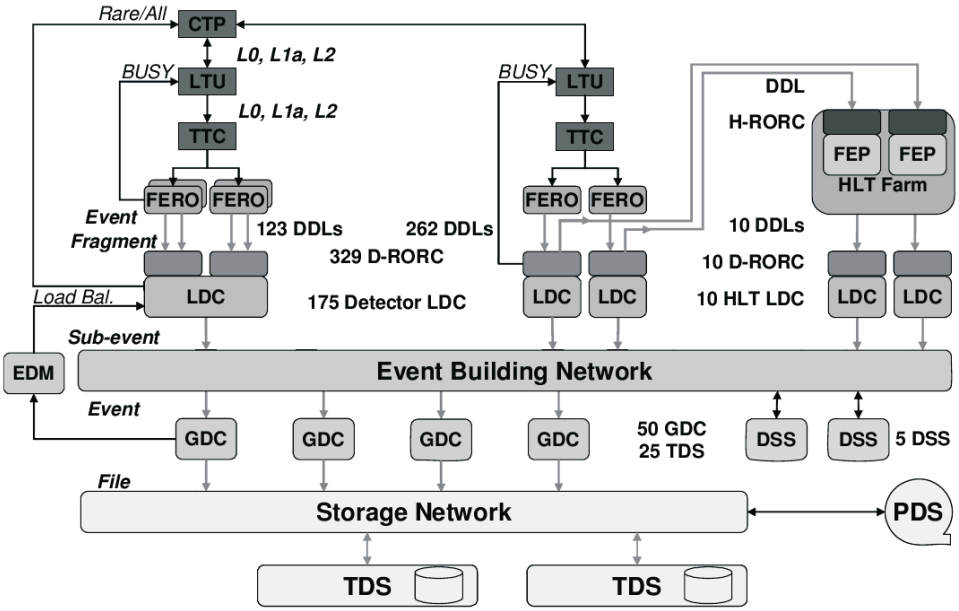


Figure 2.5: Logical overview of the Data Acquisition (DAQ) including the HLT system[8].

2.4.3 Data Acquisition (DAQ)

The role of the DAQ system in the ALICE experiment is to realize the data flow from the detector electronics up to the data storage and archiving in the CERN computing centre. The DAQ system is shown schematically in Figure 2.5. The data from the detectors are read out to Local Data Concentrators (LDCs) over the ALICE standard optical Detector Data Link (DDL). The collection of hardware read out by one DDL is denoted in ALICE as one piece of *equipment*. Each equipment in the ALICE experiment is identified by an unique *equipment ID*. This equipment ID is used during on-line or off-line reconstruction in order to retrieve the geometrical location of the equipment in ALICE.

One LDC typically receives data from 4-6 DDLs. As an example: one LDC for PHOS receives data over four DDLs from four RCUs (RCU = Readout Control Unit, see Section 4.6) so that one LDC for PHOS handles the data from one PHOS module. The data from all the LDCs of all the detectors are subsequently merged at the raw data level in the Global Data Concentrators (GDCs). The merged raw data is first written to Transient Data Storage (TDS) locally in the DAQ network in order to allow for fluctuations in the available bandwidth to Permanent Data Storage (PDS).

Subsequently the data is written to PDS.

For detectors participating in the HLT an exact copy of the data is sent to the HLT farm. The copying is done on DDL level on the Detector ReadOut Receiever Card (D-RORC) so that there is one input DDL link to the HLT for each DDL from the detector. The output data from the HLT is read back to the DAQ event building and storage network, again over the DDL. From the point of view of DAQ the HLT is read out just like any other detector.

2.4.4 The Hardware Trigger System (Trigger)

The challenge for the ALICE trigger is to make optimum use of the sub-detectors which are busy for different periods following a valid trigger. A past-future protection is applied in order to avoid pile-up. This past-future protection requirement is largest for the TPC with $\pm 88 \mu s$, due to the long drift time.

Detectors are read out in groups, denoted *clusters*. The content (sub detectors) of each cluster is programmable and is governed by the physics objective of the given run. The dead time of a given cluster is given by the slowest (i.e the largest past-future protection) detector in the cluster. The grouping is done in order to ensure that data taking for detectors with small dead time is not limited by slower ones. A detector can participate in more than one partition during the same run. As an example PHOS could participate in both a fast cluster together with for instance T0, V0 and TRD. and at the same time in cluster containing TPC and EMCAL.

Each cluster has associated with it a set of trigger classes. A trigger class is a given combination, or *signature* of triggers issued to the CTP for which the cluster should be read out.

The CTP The heart of the trigger system is the Central Trigger Processor (CTP). The CTP receives and processes the trigger signals from the trigger detectors (ITS, T0, V0, TRD, Di-muon, PHOS, EMCAL, ZDC) and then subsequently, depending on the configuration of the CTP, issues trigger to a selected set of sub-detectors (including the trigger detectors themselves).

Some detectors require the trigger to arrive at the detector after $\sim$1.2 μs. The fast part of the trigger is therefore split into two levels, L0 which reaches the detectors after $\sim$1.2 μs and a L1 trigger issued after $\sim$6.5 μs.

400 ns is needed for the trigger signals to travel to the CTP, to be processed, and to be sent back to the detector A trigger detector must therefore issue a L0 after 800 ns and a L1 after 6.1 μs. Trigger detectors fast enough to issue a L0 contributes to the L0 trigger decision and the rest to the L1 trigger decision. Some detectors, such as for instance PHOS, issues both a L0 and a L1 trigger.

The L2 trigger is issued after 88 μs. The latency requirement of the L2 trigger is governed by the past-future protection of the TPC. If the past-future protection requirement

is met (no other central event within the ± 88 μs time interval) an L2 accept (L2a) is issued, otherwise an L2 reject (L2r) is issued.

A detailed discussion of the trigger setup and of various trigger conditions is found in [8].

2.4.5 The High Level Trigger (HLT)

The data selected by the L0, L1 and L2 trigger amounts to more than 20 GByte/s (Section 2.5.1.3). Further reduction of the data rate down to the available 1.25 GByte/s bandwidth will be performed by the HLT. The HLT is discussed in more details in Chapter 3 and 8.

2.5 ALICE Running Strategy

ALICE will take its first data with p+p collisions because the LHC will be commissioned with p+p collisions, and also because elementary nucleon-nucleon collisions at the highest possible CM energy are important to the particle physics program.

In addition the p+p collisions will provide important reference data for the interpretation of results obtained from Pb+Pb collisions.

One period of p+Pb runs is required early on in order to determine the nuclear modification of the nucleonic parton distribution function (shadowing) necessary in order to interpret Pb+Pb data.

Energy density and system size dependencies will be studied using lower mass ions, for instance Ar+Ar, and heavy-ion collisions of varying centrality.

The heavy-ion programme is divided into two phases: The initial phase with priorities based on results obtained at SPS and RHIC and our current theoretical understanding. The running strategy foreseen in the initial phase is summarized below [15].

- Regular pp runs at $\sqrt{s} = 14$ TeV
- Initial heavy-ion programme

  Pb + Pb physics pilot run

  1-2 years of Pb + Pb at $\sqrt{s} = 5.5$ ATeV

  1 year pPb like collisions (pPb, dPb or αPb)

  1-2 years Ar + Ar

In addition a direct comparison between Pb+Pb and pp data will require a dedicated pp run at $\sqrt{s} = 5.5$ TeV. For the later phase, the running strategy will depend on the results from the first years of running.

Detector	p-p min. bias (kB)	Pb-Pb central (MB)
ITS Pixel		0.140
ITS Drift	1.8	1.500
TPC	2450	75.900
TRD	11.1	8.000
TOF		0.180
PHOS		0.020
HMPID		0.120
MUON		0.150
PMD		0.120
Trigger		0.120
Total	2500	86.500

Table 2.2: Anticipated event size for various sub-detectors [8].

2.5.1 Event Size and Data Rates

2.5.1.1 Event Rates

p+p Collisions Since ALICE is a dedicated heavy-ion experiment it is not suited to handle the interaction rates for p+p collisions at the full LHC luminosity. For p+p collisions the beam is therefore de-focused in the ALICE interaction area to obtain a luminosity of $\mathscr{L} = 5 \cdot 10^{30} cm^{-2} s^{-1}$ [15]. The interaction rate at this luminosity is 200 kHz for minimum bias p+p collisions. For trigger clusters including the TPC the event rate is limited by the dead-time of the TPC to about 1.2 kHz.

For other trigger clusters the event rate is limited either by the past-future protection interval of the slowest detector or by the bandwidth of the DDLs.

Pb+Pb Collisions The interaction rate for Pb+Pb collisions is expected to be ∼8 kHz. The event rate for trigger cluster including the TPC is reduced to 2 kHz by the past-future protection interval of the TPC $\pm 88 \mu s$. For the 10 % most central events this gives an event rate of 200 Hz. It is foreseen that PHOS can be read out standalone at 8 kHz using min bias Pb+Pb trigger.

2.5.1.2 Event Size

The anticipated event sizes for various ALICE sub-detectors [8] are shown in Table 2.2. For minimum bias p+p the event size is assumed to be negligible for some detectors (for PHOS, assuming that sparse readout is used).

2.5.1.3 Data Rate

Using the numbers in Table 2.2 the datarates can be calculated to be 3 GByte/s and 17.3 GByte/s for min. bias p+p and central Pb+Pb, respectively.

For PHOS, the 20 kByte event size anticipated in Table 2.2 for Pb+Pb collisions is too small. Since sparse readout will not be used for Pb+Pb collisions (Section 4.7.1.2) the minimum event size is ∼35 kByte for one module and ∼175 kByte for the whole PHOS (limited downwards by the size of the ALTRO trailers). In addition comes the data.

For cosmic data from PHOS, with an anticipated low occupancy compared to Pb+Pb collisions, the event size was found to be ∼800 kByte for data taken in 2008 (see Section 8.3.1).

For TPC and ITS the event size was also found to be larger than anticipated during commissioning at ALICE. The real data rate is therefore likely to be at least 20 GByte/s.

Chapter 3

The ALICE High Level Trigger (HLT)

3.1 Introduction

It was decided by the ALICE collaboration that the goals of the physics program could be achieved at an acceptable cost with a storage bandwidth of 1.25 GByte/s [15], [28].

The peak data rate into the front-end electronics amounts to more than 6 GByte/s. The L0, L1 and L2 hardware trigger system reduce the data read out by the DDLs which is still at least 20 GByte/s for Pb+Pb collision at nominal luminosity at 200 Hz (Section 2.5.1.3). The exact number depends on several parameters such as the configuration of the front-end electronics of participating sub-detectors, noise conditions and the configuration of the CTP.

For the sub-detectors participating in the HLT (TPC, PHOS, TRD, EMCAL, di-muon, FMD and ITS) further compression of the data will be performed by the High Level Trigger (HLT). By sophisticated on-line analysis the available storage bandwidth will be set aside for the most interesting physics events.

The overall physics requirements of the HLT can be categorized as follows.

- Compress: Reduce the event size without loss of information
- Trigger: Reject or accept events based on on-line analysis
- Select: Physics Regions Of Interest (ROI) within an event

In addition the HLT provides a natural entry point for on-line monitoring and the accumulation of calibration data to be used during the first pass of off-line reconstruction. Figure 3.1 shows the integration of the HLT into the data flow of the ALICE experiment. The raw data is transferred via optical fibers from the detector front-end to the DAQ system. The D-RORCs of the DAQ system receive the data and send two identical copies of the

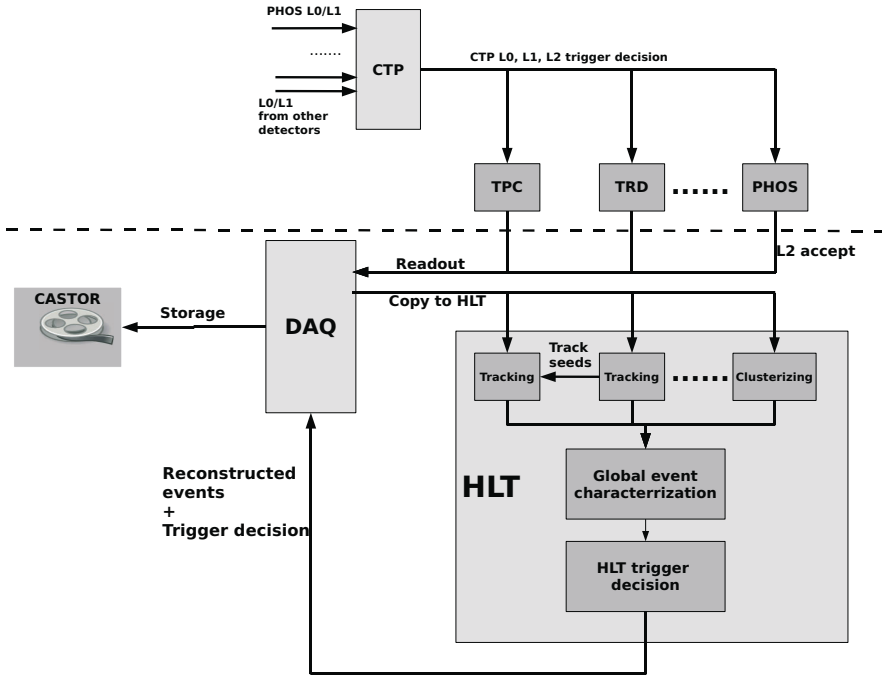


Figure 3.1: Integration of the HLT and the DAQ system. The HLT receives a copy of the detector data and is treated by DAQ, during readout, as any other detector.

data, one copy to the LDCs and the other one to the HLT. The data stream is received by the HLT Readout Receiver Cards (HLT-RORCs), which are interfaced to the Front End Processor (FEP) nodes through the internal PCI-bus. The trigger decision, reconstructed events, and compressed data are transferred back to the DAQ via the DDLs.

The different running modes of the HLT are summarized below.

- mode A: HLT ignored by DAQ (The sending of data to the HLT is disabled).
- mode B: Data and trigger decision from the HLT recorded, but the trigger decision is ignored by DAQ.
- mode C: HLT controls DAQ i.e., the HLT sends data and trigger decision which are recorded by DAQ. The busy signal and the trigger decision are respected by DAQ as indicated in Figure 3.1.

Mode C is the standard mode. In this mode the busy signal is respected by the DAQ and

the trigger system. This means that if the HLT is too busy to process an incoming trigger then the HLT sends a busy signal to DAQ which is forwarded to the CTP as indicated in Figure 2.5. Triggers are then blocked until the HLT is ready to process a new event.

Mode B is similar to mode C except that the raw data from the sub-detectors are recorded while the trigger and RIO (see Section 3.3) are ignored by DAQ.

3.2 Compression

The data compression is achieved at two levels, at the first level by extracting basic signal parameters from the raw data. For PHOS this comprises the extraction of the amplitude and the time-of-flight from the over sampled time dependent signal as described in Section 6.2.1.

The second level of compression is applied upon the reconstructed event. Various techniques have been considered including common audio and video compression techniques such as MPEG and JPEG. More details can be found in [8]. In addition the use of entropy encoding has been considered for PHOS [41].

3.3 Trigger and Region of Interest (ROI)

The output of the HLT on-line processing for a given event can be divided in three parts:

- Event Summary Data (ESD).
- Trigger information i.e data describing which physics condition lead to a HLT trigger.
- ROI telling DAQ for which equipments the full raw data should be recorded (a list of DDLs/equipments).

The ESDs and the trigger information is contained in the HLT output data. Since DAQ does not read or analyze the content of the data received from any detector, including HLT, the trigger decision from HLT is therefore from the point of view of DAQ equivalent with the ROI. If the event was rejected by the HLT then ROI will be an empty list of DDLs. For ordinary events the ROI will typically specify that only the HLT should be read out. For rare events the ROI will specify, in addition to the readout of the HLT, raw data to be read out from parts, or *regions*, of the detector containing *interesting* signals. Thereby the name Region Of Interest.

The ROI might of course also specify that the full raw data from all detectors should be recorded. This is equivalent to running the HLT in mode B.

3.3.1 Trigger Scenarios

Since no collisions have yet been produced at the LHC, the HLT was not used as a physics trigger so far. The HLT physics trigger is currently in the implementation phase. Trigger scenarios listed in this section are based on [8] and on recent internal discussions in HLT and PHOS.

- Jet physics

  High p_T jets: Triggering on jets with high (50 - 200 GeV/c) transverse momentum using EMCAL, TRD, ITS and TPC

  γ - jet correlation: Same as above with an additional trigger requirement that an *isolated* high energy photon hits PHOS (see Section 4.2.3).

- Ultra peripheral collisions: Improved statistics for ultra-peripheral events can be achieved by triggering on the condition that exactly two tracks with opposite charge emerging from the primary vertex are observed in the TPC.

- Open charm detection using the $D^0 \rightarrow K\pi$ decay channel: approximately one month of central Pb+Pb running at 20 Hz is needed. If all events are written to storage up to 850 MByte/s is needed for this observable alone. Only about one in 1000 of the events are anticipated to contain a detectable D^0 decay. The signal to noise can therefore be improved by three orders of magnitude by detecting this decay on-line.

- Dilepton physics

  Dielectron trigger: The J/ψ and Υ can be identified by their decays into e^+e^- pairs by tracking through the TPC, TRD and ITS. The HLT will reduce the event rate by a factor $\sim$10 by applying an invariant mass window around the J/ψ and Υ family.

  Dimuon trigger: The main task of the HLT for the muon arm is to refine the momentum cut for the J/ψ and Υ detection. There are two cuts: One low threshold $p_T > 1$ GeV/c for the J/ψ and a high threshold $p_T > 2$ GeV/c for the Υ resonance. Simulation studies [8] have shown that the signal to background can be improved by the HLT by a factor five for J/ψ and by a factor 100 for Υ resonances.

3.4 Calibration

It was decided that the sub-detectors should develop software algorithms, or Detector Algorithms (DA), to accumulate calibration data to be used during the first pass of off-line reconstruction. These algorithms should run either in the DAQ system, in the HLT, or off-line. Furthermore, the algorithmic part of the DA should be identical in all three cases.

3.5 Hardware Architecture

The HLT will consist of up to ∼1000 PCs. The cluster nodes can be divided in two categories.

- Front End Processor (FEP): A node receiving data directly from the detector via the DDLs and the H-RORC, each FEP receiving data from four DDLs.
- Computing node (cn): a node receiving data from one or more other nodes.

Each FEP is equipped with two custom build PCI-X based readout receiver cards (H-Rorc) each receiving detector data from two DDLs. Each H-Rorc is equipped with a powerful FPGA co-processor (Xilinx Virtex4 LX40).

3.6 Software Architecture

The processing of the data follows the natural hierarchy and distribution of the readout partitions of the detectors. The data processing is carried out by individual software components running on the nodes of the computing cluster. The data is pushed through the components during the analysis where each component represents a different level of the processing. In the first step of the processing one event is processed in parallel distributed over the nodes. This concept follows the optimization of the HLT to reduce copying of data to a minimum. In the subsequent analysis layers the output from the first step of the analysis is merged for global event reconstruction. In the last step of the analysis, a trigger decision is made based on the reconstructed event and the specific trigger criteria. Data reduction is mainly achieved by partial detector read-out, intelligent event selection, and data compression algorithms. Further HLT applications include short event summary data (ESD), on-line detector monitoring, and calculation of calibration data.

The HLT software is divided into two functional parts, the data transportation framework and the data analysis. The first part contains the communication of the components, steering, and data transfer between components and nodes. The second part contains the physics analysis. The two parts communicate via well defined interfaces. This concept ensures:

- A high degree of flexibility.
  Development of Analysis Components is independent from the data transportation.
- Quality of the analysis.
  Analysis Components can be run in the off-line analysis without changes. The results are directly comparable.

- Clear development framework and policies.
  Developers of Physics Analysis Components can use an environment they are familiar with.

- Reproducibility of trigger decisions.

3.6.1 Data Transportation Framework

The HLT system has been designed to reduce the amount of data to be copied to a minimum. The entire communication mechanism is designed for a low processing overhead. Figure 3.2 shows the working principle of the on-line framework. Based on the publish/subscribe paradigm, a generic data transportation and steering framework has been developed, denoted the *HLT PubSub* framework. It carries out the task of communication, data transportation and load distribution within the HLT system. Data processing is based on individual components running on the different cluster nodes and interfacing to the detector algorithms, the actual physics analysis part. All data exchange and communication between components is performed transparently for the developer by the PubSub framework.

The components communicate via *named pipes* and exchange small data descriptors specifying, among other things, where the output data produced by one component can be found by another *receiving* component (typically an address of a block of shared memory).

On one node, the data itself is exchanged via shared memory using the data descriptor to locate the data, thereby avoiding unnecessary copying of data. The communication of data is performed transparently for the user applications. More details about the on-line framework can be found in [9].

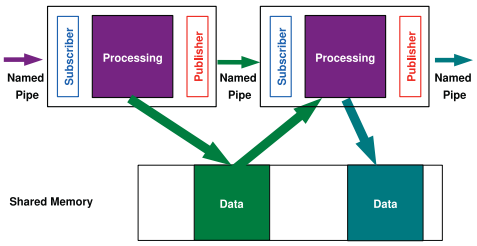


Figure 3.2: Working principle of the *HLT Publisher Subscriber* framework [9].

An HLT analysis chain can run in either off-line or on-line mode. Data is provided to the user applications by means of two dedicated software components which are a

part of the framework, the *FilePublisher*[1] used to provide data in off-line mode and the *RorcPublisher* for providing data in on-line mode. In on-line mode the data is received directly from the detector via the DDLs. In off-line mode the data is read from file and distributed over the nodes [9].

The detector specific analysis software sends its output data to a *HLTOUT* node which handles the communication between HLT and DAQ.

3.6.1.1 Monitoring

On.line access to data is made available at any level of the processing by the software package HOMER (HLT On-line Monitoring Environment including ROOT) which is an integrated part of the HLT framework. The HLT on-line monitoring for PHOS (Section 8.5) is based on this package.

The software component used for PHOS for monitoring purpose is denoted the *TCPDumpSubscriber*. This component can subscribe to the output of any processing component running in the HLT and sends this output to the detector specific monitoring tool over TCP/IP.

3.6.2 On-line Data Analysis

Data Analysis in ALICE is done with the analysis framework *AliRoot* which contains code for simulation of events and detector response as well as the event reconstruction and physics analysis. The simulation and reconstruction is split into modules, one for each detector. In that framework, the HLT acts as an additional detector.

The analysis code is binary exactly the same in on-line and off-line mode.

This concept of running exactly the same code on-line and off-line gives an unique possibility to perform valid tests off-line. This feature has been heavily used during the development of the PHOS HLT software (Section 8).

ALICE HLT's approach and the framework allow to distribute single events over the cluster nodes. Splitting of the processing reduces the amount of data to be copied dramatically as the first step of the reconstruction can be performed already on the *FEPs*. The distribution of the event processing is most effective on the FEP nodes. Between the FEP nodes and the computing nodes, copying of data is unavoidable. It is therefore desirable that the data is compressed as much as possible at the FEP nodes before it is sent to the computing nodes.

For PHOS, the energy deposited in the detection channels and the time of flight from the interaction point are evaluated, using the Peak-Finder algorithm, as described in Section G.1, thereby reducing the amount of data by a factor 10 - 20.

[1] There are actually several file publishers available in the framework, but they perform logically the same task.

Part II

Commissioning of the ALICE PHOS Detector and Integration into the ALICE High Level Trigger

Chapter 4

The Photon Spectrometer (PHOS)

4.1 Introduction

The role of PHOS in the ALICE experiment is to detect and identify photons and measure with high resolution their 4-momentum with the prospect of performing the ALICE physics programme exploiting direct photons and light neutral mesons (π^0 and η mesons mainly) as probes [15].

PHOS is a high resolution electromagnetic calorimeter with a dynamic range of 5 MeV - 100 GeV covering 100° in azimuthal angle and $|\eta| < 0.12$ in pseudorapidity, and is situated 4.6 m from the interaction point.

PHOS consists of two detectors, an electromagnetic calorimeter (EMC) and a Charged Particle Veto (CPV) detector. The calorimeter consists of lead tungstate crystals ($PbW0_4$) with a high granularity and high energy resolution which are read out by APDs. The CPV consist of multi-wire proportional chambers with one CPV module for each EMC module. The CPV will provide discrimination against charged particles. At the time of writing it is not clear if when the CPV will be ready. The EMC is therefore the main focus in this chapter.

4.2 Photon Physics with PHOS

Both direct photons and light neutral mesons decaying into diphotons are important observables for the ALICE physics programme, direct photons as penetrating probes carrying unaltered information from early collision stages (see Section 1.3), and light neutral mesons (π^0 and η mesons mainly) as probes of jet quenching. The latter topic will be studied both through nuclear modification of single-particle meson spectra and via γ-jet and γ-hadron correlation, using the TPC and EMCAL for jet detection and PHOS and EMCAL for detecting the trigger photon.

The detection of photons in PHOS is straightforward. The main challenge is to discriminate between photons and the abundance of other particles. Light neutral mesons are measured by invariant mass calculation using their $\gamma\gamma$ decay channel. A good energy resolution and high granularity is necessary in order to obtain a good resolution in the two photons invariant mass spectrum up to ∼30 GeV/c. Above 30 GeV/c the two photons from a π^0 decay can no longer be separated, and shower topology analysis is used instead.

The high granularity also facilitates the rejection of hadronic showers which is also performed using shower topology analysis (Chapter 6). $\bar{n}n$ annihilation can only be distinguished from photons using time-of-flight, thereby requiring a good time-of-flight resolution. Simulation studies[15] have shown that a time-of-flight resolution of 1 ns at ∼2 GeV is sufficient.

4.2.1 Extraction of Direct Photon Spectrum

At low to intermediate p_T, the direct photon spectrum must be determined on a statistical basis, using the measured inclusive photon spectrum and the spectra of electromagnetically decaying neutral mesons (see next subsection). At high p_T ($\geq$20 GeV/c) direct photons might be distinguished on an event-to-event basis. Two different techniques may be used, a shower shape analysis to discriminate between single photons and π^0s forming overlapping clusters, and an isolation cut, labelling a photon as prompt if it appears without hadrons travelling in the same direction.

4.2.2 Direct Photons and Light Neutral Meson Spectra

The main source of systematic errors in the direct photon spectra is the subtraction of light neutral mesons, mainly π^0 and η. Accurate determination of the π^0 and η spectra, from their diphotonic decays, is crucial in order to extract the direct photon part but these mesons are also interesting probes themselves. If a QGP is present the light mesons will be suppressed, since they are formed by fragmentation of partons which has interacted strongly with the QGP.

The two photon invariant mass is calculated for particles identified as photons (Section 6.5). The two photon invariant mass is given by the formula

$$m\gamma_1\gamma_2 = \sqrt{2E_{\gamma_1}E_{\gamma_2}(1-\cos\theta)} \tag{4.1}$$

where $m_{\gamma_1\gamma_2}$ is the invariant mass, $E_{\gamma 1}$ and $E_{\gamma 2}$ is the energy deposited by the two photons and θ is the opening angle between them. The mass resolution is given by the formula

$$\frac{\sigma_{m\gamma_1\gamma_2}}{M} = \sqrt{\frac{\sigma^2_{E_{\gamma_1}}}{E^2_{\gamma_1}} + \frac{\sigma^2_{E_{\gamma_2}}}{E^2_{\gamma_2}} + \frac{\sigma^2_\theta}{\tan^2(\theta/2)}} \tag{4.2}$$

As seen from Equation 4.2 the mass resolution depends on both the energy resolution and the position resolution. The dependence on position enters through the opening angle θ which is measured using the geometrical position of the two hits as given by Equation 6.13, assuming that both photons originate from the primary vertex.

In addition the mass resolution depends indirectly on the time-of-flight resolution since the time-of-flight is one of the criteria used to discriminate photons from other particles (see Section 6.5).

Acceptance "The acceptance for π^0 and η mesons detected in the $\gamma\gamma$ decays channel is defined as the probability that both decay photons hit PHOS when the mesons are radiated from the interaction point within rapidity interval $|y| < 0.5$ and azimuth angle $0 < \varphi < 2\pi$ [28]." At low p_T the acceptance is limited by increasing opening angles between the two decay photons leading to a decrease of the acceptance when the energy of the initial meson is below a few hundred MeV. The acceptance peaks at around $\sim$20 GeV/c. At higher energies the probability that the two decay photons form a single cluster in PHOS increases, leading to a sharp decrease in the acceptance above $\sim$30 GeV/c.

Combinatorial Background The combinatorial background arises from photon pairs accidentally having an invariant mass similar to that of a π^0 or η meson, but which are not decay photons from these mesons.

The combinatorial background will be determined in PHOS using the so called *event mixing* technique. The two photon invariant mass distribution will be evaluated for pairs of photons from different events having similar global characteristics (primary vertex, centrality etc.).

4.2.3 γ - jet Correlation

During initial hard processes back to back jets can be formed with a with a prompt photon in one direction and a parton in the other, the latter finally fragmenting into a hadron jet.

The initial total momentum is zero in the transverse direction. If a QGP is present, any energy loss of the jet can be measured by detecting the photon.

Since the photon does not interact with the QGP it should be *isolated*, i.e. not surrounded by hadrons emitted in the same direction. However, at least in heavy-ion collisions, this topology will be perturbed by the underlying event. An additional challenge is that the cross section for gamma production is much lower than for hadrons.

In order to improve the statistics for γ - jet correlation studies it is therefore foreseen that the PHOS HLT should provide an isolated photon trigger (see Chapter 8).

4.3 Principle of Operation

4.3.1 Shower Development in the PHOS Crystals

Electromagnetic Shower An electromagnetic shower is created when a high energy photon, e^- or e^+ hits a crystal. The shower development is similar in all three cases and will only be described in the case of a high energy photon below. A typical example of an electromagnetic shower is shown in Figure 6.4.

When a high energy photon interacts with the crystal it is absorbed and a e^+e^- pair is produced. These leptons will then emit bremsstrahlung photons giving rise to more e^+e^- pairs. Hence the energy loss is mainly through e^+e^- pair production. This gives a shower of e^+e^- pairs and photons.

The radiation length, L_r, is defined as the distance an electron or positron has to travel in order to loose $1/e$ of its energy by bremsstrahlung. It is shown in Appendix B that $L_r = 0.89$ cm for PHOS. The mean free path for pair production is $7/9L_r$. The propagation of the shower is mainly in the longitudinal direction relative to the trajectory of the incident particle, but there is also some propagation in the transverse direction plane so that the shower will develop as a cone. The lateral dispersion of the cone is given by the Moliere radius which for PHOS is ~ 2 cm (see Appendix B).

The shower develops until the energy of the secondary e^+ e^-s falls below a critical energy, E_C, where the ionization loss equals that of bremsstrahlung. The critical energy E_C is approximately

$$E_C = \frac{610MeV}{Z + 1.24} \tag{4.3}$$

for a single material. For a compound the Bragg's rule can be used (see Appendix B).

A simplified but still quantitatively correct model of the electromagnetic shower is obtained by assuming that a photon with an initial energy E_0 is absorbed by creating an e^+ e^- pair with exactly half of the photon energy each. In this idealized model the secondary e^+ e^- give up exactly half their energy, while the photons are transformed into e^+e^- pairs sharing the γ-energy equally after traveling exactly one radiation length [42].

If $E_0 >> E_C$ then the shower will contain 2^t particles after t radiation lengths, each particle with an average energy of

$$E(t) = \frac{E_0}{2^t} \tag{4.4}$$

The shower will stop developing when $E(t) = E_C$ and $t = t_{max} \equiv ln(E_0/E_c)/ln2$. The number of particles at this point is

$$N_{max} = \exp(t_{max} ln2) = E_0/E_C \tag{4.5}$$

with equal numbers of electrons, positrons and photons.

The electrons and photons formed in the pair production process excite the atoms of the crystals which then scintillate. This scintillation light is measured by the APD.

If the critical energy E_C is reached before the secondary electrons have traversed the length of the crystal then the number of electrons positron pairs, and thereby the amount of scintillation light, is proportional to the energy of the primary particle and the calorimeter is said to be linear in this region. For PHOS, the length of the crystals is chosen in order to give a linearity better than one percent in the range 20 MeV to 100 GeV.

Hadronic Shower In a high energy heavy-ion environment hadrons enters PHOS mainly around MIP energies. A large fraction of the hadrons can therefore be discarded by applying a selection cut on the energy after clusterization of ∼200 MeV prior to the analysis [14].

The probability that a hadron creates a hadronic shower in PHOS depends on the particle type and the energy. The development of the shower is much more complicated than for an electromagnetic one since both electromagnetic and hard inelastic scatterings are involved. Depending on the proportion of π^0 mesons created in the early stage of the shower the shower might develop either as an hadronic or an electromagnetic one. Thus there are large fluctuations in the shower profile. Figure 6.5 shows an example of a typical hadronic shower.

Neutral hadrons will be identified in PHOS by measuring the lateral dispersion of the shower (Section 6.5). For identification of charged hadrons one will in addition use projection of the CPV hits on the PHOS crystal plane [14].

4.3.2 Properties of the $PbWO_4$ Crystal

Some of the properties of the $PbWO_4$ are given in Table 4.1 and discussed further in Appendix B. The small radiation length combined with a good transparency (low self absorption) of the scintillation light makes the crystal well suited to measure signals over a large dynamic range. In addition the small Moliere radius facilitates good position resolution needed for two-photon separation and invariant mass calculations. The scin-

Density	8.28 g/cm^3
Radiation length	0.89 cm
Interaction length	20.7 cm
Moliere radius	2.0 cm
Decay time (fast/slow)	10/30 ns
Light Yield	0.3 %
Melting point	1123 $°C$

Table 4.1: Some properties of the PWO crystal [14].

tillation light is composed of a fast and a slow component with a decay time of 10/30 ns

respectively. This is very fast compared with most other materials giving a good intrinsic time-of-flight resolution in the order of 500 *ps* at 2 GeV.

On the negative side the light yield is relatively low which makes it necessary to multiply the scintillation photons prior to further processing. This multiplication is performed by the APD.

4.3.3 The Avalanche Photo Diode (APD)

An Avalanche Photo Diode (APD) is the semiconductor analog to a photomultiplier. The APD is a photo diode operated with a reverse bias voltage typically in the range 100 - 400 V. Hole-electron pairs are created by absorbed photons creating a current of primary electrons.

If the APD is operated close to its breakdown voltage the electrons gain enough energy to create additional hole-electron pairs, producing an avalanche multiplication effect.

The gain, M, is defined as the number of electrons after the avalanche multiplication divided by the number of primary electrons.

There is a statistical error induced by the multiplication process since not all primary electrons obtain exactly the same multiplication. This error is denoted *excessive noise* and increases linearly with M for large value of M. This limits the applicable gain to typically no more than 50 - 100.

One advantage of the APD is that the Quantum Efficiency (Q.E.) is very good. For the Hamamatsu S8148/S8664-55 used by PHOS the Q.E is 60 - 80 %. Another advantage is that the response time is very fast, compared to for instance a photo-multiplier.

4.3.4 The Charge Sensitive Pre-amplifier (CSP)

The CSP integrates the current from the APD and gives as an output a voltage pulse with a short rise-time of $\sim$10 ns and a long fall time of $\sim 130\mu s$ which is approximately a step pulse. The amplitude of this step is proportional to the number of electrons in the integrated current and therefore proportional to the energy deposited by the incident particle.

4.3.5 Signal Chain

In order to make the signal suitable for digitization with the ALTRO chip the output signal from the CSP is shaped using a shaper with semi-Gaussian step response in the time domain given by Equation 4.13.

Figure 4.2 shows an abstraction of the signal chain for one FEEC.

Figure 4.3 shows a typical PHOS signal for a single readout channel after digitization with the ALTRO chip. The energy deposited in the crystal is proportional to the amplitude after the subtraction of the baseline. The start of the signal, t_0, and the peak

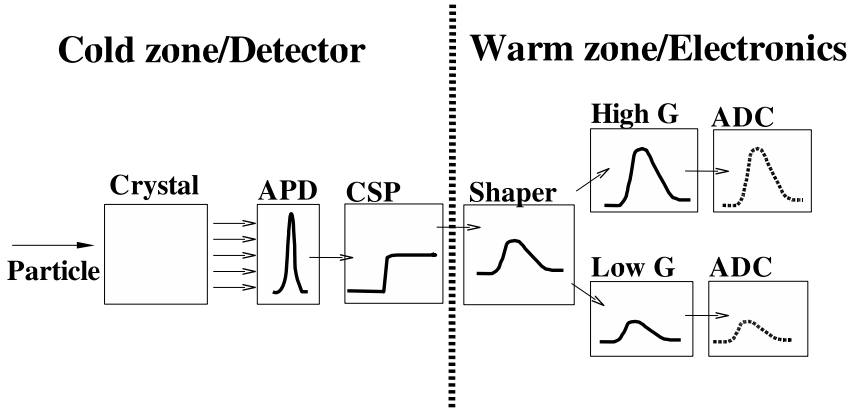


Figure 4.1: Signal chain including the digitization (sampling) by the ALTRO. From left to right: A particle hits the crystal and a shower of scintillation light is created. The scintillation light is absorbed by the APD which converts the light to a current pulse. The current pulse from the APD is integrated by the CSP giving a step voltage signal. The step signal is conditioned by the shaper with two different gains and made suitable for sampling/digitization by the ALTRO chip.

position, $t_0 + \tau$ is linearly dependent on the time-of-flight from the collision point to PHOS.

4.4 Factors Affecting the Energy, Time of Flight and Position Resolution

4.4.1 Factors Affecting the Energy Resolution

The energy resolution for calorimeters can in general be parametrized as [28] [14]

$$\frac{\sigma_E}{E} = \sqrt{\left(\frac{a}{E}\right)^2 + \left(\frac{b}{\sqrt{E}}\right)^2 + c^2} \tag{4.6}$$

where E is in units of GeV.

a is mainly due to electronic noise, which is assumed to be independent of the energy of the detected particle. This is the dominating term at low energies[14].

The stochastic term b represents statistics related fluctuations such as intrinsic shower fluctuation, photo electron statistics and dead material in front of the detector. This term dominates in the intermediate energy range ($\sim$ 5-10 GeV). The stochastic shower

fluctuations are smaller, on a relative scale for a higher light yield of the $PbWO_4$ crystal. For this reason a lot of R&D has been undertaken in order to improve the light yield of the crystals. For the same reason the volume containing the crystal is cooled down to $-25°$C where the light yield is three times higher than at room temperature [14].

The constant term c, takes into account shower leakage at the back end of the crystals, imperfect channel to channel relative calibration, and geometrical effects. This constant term becomes very important at very high energies[14]. The magnitude of this term can be reduced by ensuring that the inter-calibration is as good as possible and by taking non-linearity effects into account during the analysis.

4.4.2 Factors Affecting the time-of-flight Resolution

Time-of-flight measurements are most relevant for energies around $\sim$2 GeV in order to discriminate between $n\bar{n}$ annihilation (which gives a signal around sim 2 GeV) and a true γ signal.

The time-of-flight resolution is limited downwards by the intrinsic time-of-flight resolution of 500 ps of the $PbWO_4$ crystal. Other factors that influences the time-of-flight resolution is the charge collection time of the CSP, the electronic noise and the time constant of the shaping circuit. It can be shown that the time-of-flight resolution is inversely proportional to the energy and proportional to the square root of the time constant of the shaper [43].

4.4.3 Factors Affecting the Position Resolution

The two most important factors that determine the position resolution are the Moliere radius and the granularity of the detector. The relatively small Moliere radius of 2 cm was one of the main reasons why the $PbWO_4$ was chosen as detector material for PHOS. The 2.2x2.2x18 cm^3 dimension of the crystals are chosen in order to match the Moliere radius.

4.5 Physical Layout

PHOS consists of 17920 detection channels of lead tungstate crystals ($PbWO_4$) segmented in 5 modules of 64x56 crystals each. Each crystal is equipped with an Avalanche Photo Diode[1] (APD) as photo detector and a low noise JFET Charge Sensitive Preamplifier (CSP). The APD is glued to the end of the crystal, and the CSP is mounted to the back end of the APD. The PHOS module is divided into a *warm* and a *cold* volume. The crystals, APDs and the CSPs are placed inside the cold volume at -25 ± 0.3 ^{0}C. The

[1]Hamamatsu S8664-55

Component	Detectors
DCS (board)	PHOS, EMCAL, TPC, FMD, TRD
RCU	PHOS, EMCAL, TPC, FMD
FEEC (PHOS type)	PHOS, EMCAL
TRU	PHOS, EMCAL
TOR	PHOS, EMCAL
Busy Box (BB)	PHOS, EMCAL, TPC, FMD
ALTRO	PHOS, EMCAL, TPC, FMD
BC	PHOS, EMCAL, TPC, FMD
SIU + DIU	All detectors

Table 4.2: Hardware components and devices used for PHOS and their use by other detectors.

readout electronics is situated in the warm zone where the temperature is regulated to approximately room temperature.

One detection channel for PHOS consists of a 2.2x2.2x18 cm^3 $PbWO_4$ crystal, one APD and one CSP. The crystals are mounted in strip units of eight crystals. The signals from 4 strip units are connected to a Front End Electronic Card (FEEC) in the "warm zone" through two feed-through connectors as shown in Figure 4.4(f). The analog shapers on the FEEC give a high gain and a low gain output for each crystal for a total of 64 outputs from the 4x8=32 crystals. These 64 signals are digitized with 4 ALTRO digitizer chips [44], each with 16 ADCs.

The LED Monitoring System The detector response is monitored by flashing the APDs with light with a wavelength distribution similar to the scintillation light from the crystals. The LED monitoring system is situated inside the cold volume and has one LED for each crystal.

4.6 Front-End Electronics

Table 4.2 shows the hardware devices that are used for PHOS and their use for other sub-detectors. For configuration and monitoring, the access to all the components are done directly or indirectly via the DCS board which is described in further detail in Section 4.6.4.

One RCU reads out 2x14 FEECs and four RCUs read out one PHOS module as shown in Figure 4.11.

From the point of view of software development on the DCS side there are three different types of devices that has to be accessed and controlled: The ALTRO or ALTRO like

devices, the BC and the RCU or RCU like devices. The various *DCS nodes* or *DCS flavors* are discussed in Section 4.6.4.

4.6.1 The PHOS FEE Card (FEEC)

The PHOS Front End Electronics readout Card is shown in Figure 4.7. One FEEC reads out two strip units i.e 2x16 detection channels. The signals arriving from the detection channels are shaped with 2 μs semi Gaussian shapers. A detailed treatment of the PHOS shaper and the detector response can be found in [43] and is only repeated briefly here.

The frequency domain step response of the *nth* semi Gaussian shaper is

$$H(s)_{SG} = \Big[\frac{s\tau_0}{1+s\tau_0}\Big]\Big[\frac{A}{1+s\tau_0}\Big]^n. \tag{4.7}$$

where s = $j\omega$ is the complex angular frequency, τ_0 the time constant of the differentiation and the integration stages of the shaper, which for PHOS is $\sim 1\ \mu$ s, and A is the amplitude of the signal.

For the second order shaper used in PHOS the transfer function is

$$H(s)^{n=2}_{shaper} = (s+p_z)\frac{s\tau_0 A^2}{(1+s\tau_0)^3} \tag{4.8}$$

The pole zero cancellation term $(s+p_z)$ is included to account for the fact that the output of the CSP is not a perfect step function. The fast rise time of the CSP affects only frequencies much higher than the sampling frequency of the ALTRO and does not significantly affect the signal seen after digitization. The CSP impedance can therefore be approximated by

$$C(s) = \frac{\tau_{RC}}{1+s\tau_{RC}} \tag{4.9}$$

where τ_{RC} is the time constant of the RC feedback circuit of the CSP with R = $100M\Omega$ and C = 1.3 pF giving $\tau_{RC} = 130\ \mu s$. Ideally the time constant of the pole zero cancellation term should be equal to the time constant τ_{RC} so that the zero of $(s+p_z)$ and pole of C(s) cancel i.e. $p_z/s = \tau_{RC}$, hence the name pole zero cancellation. If the pole of 4.9 and zero of 4.8 cancel each other, one is left with the an effective system function of the CSP an shaper given by

$$H(s)_{CSP+shaper} = \frac{s\tau_0 A^2}{(1+s\tau_0)^3} \tag{4.10}$$

in the frequency domain. In the time domain this translates to

$$f(t) = A^2(\frac{t}{\tau_{shaper}})^2 \exp(-2\frac{t}{\tau_{shaper}}), \tag{4.11}$$

where t is the time, A is the amplitude and τ_{shaper} is the rise time, i.e. the time between the start of the time dependent signal and the peak position. An example of a signal on

the form given by Equation 4.11 is shown in Figure 4.3. For a given channel, τ_{shaper} is a constant, however, this constant can vary slightly from channel to channel. For an *nth* order shaper, τ_{shaper} is related to the time constant of the differentiation stage and the n integration stages (which have the same time constant) by the relation

$$\tau_{shaper} = n\tau_0 \tag{4.12}$$

For PHOS, with $\tau_0 = 1\mu s$, the time constant of the signal is $\tau_{shaper} = 2\ \mu s$. During the discussion in Chapter 6 it is useful to replace t with $t - t_0$ to obtain

$$f(t) = A^2(\frac{t - t_0}{\tau_{shaper}})^2 \exp(-2\frac{t - t_0}{\tau_{shaper}}) \tag{4.13}$$

Figure 4.6 shows a Bode plot of the frequency response of the shaper assuming perfect pole zero cancellation. The dotted blue line is the response of the differentiation step. The solid red line is the response of the second order integration stage and the stitched red line the total response of the shaper. As seen in the figure the -20 db cutoff frequency is $\sim$1 MHz. The peak frequency of the bandpass filter is $\sim$ 150 KHz. The shaper circuit has two different amplification gains, low gain and high gain, in order to cover the large dynamic range so that there are 64 readout channels for each FEEC. The high gain channel covers the range 5MeV to 5 GeV whereas the low gain channels covers the range 80 MeV to 80 GeV. The digitization is done with the Alice TPC Readout Chip (ALTRO). Each ALTRO has 16 ADC inputs so that four ALTRO chips are needed for the readout of one FEEC.

Each PHOS FEEC also delivers eight fast analogue sums of 2x2 detection channels to the TRU. The PHOS trigger electronics incorporating the Trigger Region Unit (TRU) and the Trigger OR (TOR) is described in Section 4.6.2. Referring to Figure 4.7 there are two ALTROs on each side of the one FEEC. The documentation on the ALTRO chip is found in [45].

4.6.1.1 Comments on the Shaper Option Chosen for PHOS

The semi Gaussian shaper has its name from the fact that as one adds more RC integration stages the step response approaches a Gaussian function. A theoretical 18% reduction of the electronic noise can be achieved if the step response is a Gaussian function as compared to a simple CR-RC shaping circuit, however two integration stages are not enough. At least 4-5 stages are needed [46].

The electronic noise depends on many parameters but in a detector readout system, many of these parameters, such as for instance the detector capacitance, are fixed. For PHOS the properties of the detection channel i.e the crystal, APD and CSP could not have been be modified at the time the shaper was developed. The only parameter that could have been tuned in order to satisfy the energy and time-of-flight requirements was the time constant of the differential input stage of the shaper.

For a given optimum CR time constant the second order shaper has twice as long rise time as a first order shaper according to Equation 4.12. In PHOS one wants to extract both energy and time-of-flight information from the same digitized signal. A first order shaper would clearly have been the best compromise between time-of-flight and energy resolution requirements. The time-of-flight resolution will be degraded by a factor $\sim$2 for a second order shaper as compared to a first order shaper. A factor $1/\sqrt{2}$ comes from the $1/\sqrt{\tau}$ dependence of the time-of-flight resolution [43]. Another factor $1/\sqrt{2}$ comes from the fact that one is oversampling and that the cutoff frequency of the shaper is inversely proportional to the rise time, which can easily be seen from Equation 4.7. The effect is that the *richness* of the data increases by a factor two for a first order shaper compared to a second order shaper thereby increasing the effective statistics by a factor $\sqrt{2}$. The argument is quite simply that if the cutoff frequency is halved, then the Nyquist frequency is doubled.

An additional disadvantage is immediately seen from the triple pole in the denominator of Equation 4.10. Each pole is implemented with physically different components with actual values that can differ up to 5% for resistors and 10% for capacitors. It can easily be shown that for small discrepancies between the nominal and actual values of the components the triple pole becomes either two complex conjugated and one real pole or three different real poles. The right side of Equation 4.10 can then be separated in three terms to give a sum of three exponentials in the time domain. Fitting the real signals with Equation 4.10 in this case will give a systematic error in the estimate of A and t_0 since the actual analytical form is different. This systematic error can be substantial as shown in Figure G.5.

The advantage of the Semi Gaussian signal shape on the form given in Equation 4.13 is that the only free parameters A and t_0 are exactly the information one wants to extract from the raw data, therefore simplifying the analysis.

4.6.1.2 The PHOS Board Controller (BC)

The Board Controller (BC) has to perform the following tasks [10].

- Monitor the health status of the board reading current levels, voltage levels and temperatures on the FEEC.
- Notify the RCU via a dedicated interrupt line if any monitored values violate a configurable threshold.
- Control the ALTRO bus signals and the GTL drivers used in the communication network with the RCU.
- Enable the Sparse Readout functionality of the FEEC, i.e. enable the RCU to read only channels where data has been buffered.

- Configure the DACs that set the High Voltage bias voltage to the APDs.

APD Bias Control The APD bias control from the hardware point of view is shown in Figure 4.8. The board controller contains dedicated registers, one for each detection channel, for DAC values used to control the HV applied to the APDs. The DAC values are written to the BC during the configuration phase (see Chapter 5) and subsequently transferred to the DACs by issuing a command to the BC from DCS. The actual transfer of the DAC settings from the BC to the DACs is performed by the BC firmware transparently to the user. A complete documentation of the PHOS BC firmware can be found in [10].

The output from the DACs are controlled by 10 bit values in the range 0 - 5 V. There are four DACs on each FEEC, each having eight outputs to control the HV applied to the APDs of 32 detection channels. The analogue low voltage from the DAC is converted to a high voltage linearly in the the range 210 - 400 V by the LV to HV converter where a DAC code of zero gives HV output of 210 V and a DAC code of 1023 gives an output of 400 V. A more detailed description can be found in [11]

4.6.1.3 Hardware Interlock Thresholds

Each temperature, voltage or current sensor has associated with it a legal range. i.e. a minimum and a maximum value. If the value read from the sensor is outside this range, and interrupts are enabled, an interrupt is issued to the RCU over a dedicated interrupt line on the GTL bus.

4.6.2 Trigger Electronics

Based on experience from similar experiments, for instance PHENIX at RHIC, it was decided that PHOS should also be a trigger detector issuing both a L0 and L1 trigger. This requirement is rather new, from around 2003. It was not foreseen in the original PHOS TDR. A a consequence there are limited simulation studies and explicit requirements in terms of trigger efficiencies etc in [15] and [28]. The main purpose of the PHOS trigger is to improve the trigger efficiency for high p_T events in PHOS i.e. to make PHOS self triggered, thereby extending the capability of PHOS to collect statistics of photons up to a higher p_T. The PHOS L0/L1 is issued to the CTP like any L0/L1 trigger from any of the detectors in ALICE and can also be used to trigger other sub-detectors.

4.6.2.1 The Trigger Region Unit (TRU)

At the time of writing, the design of the TRU firmware is not completely finished. Referring to Figure 4.11 there are eight TRUs per PHOS module, one for each branch of 14 front-end cards. Each FEEC delivers eight 2x2 analogue sums of neighbouring channels to the TRU over a dedicated flat-band cable so that one TRU receives 112 analogue 2x2 sums

over 14 cables. The TRU searches for 2x2 sliding window sums of the 2x2 input signals (i.e 4x4 sums of physical detection channels) to produce 91 sliding window sums. These sums are compared to three different programmable threshold: Low p_T, intermediate p_T and high p_T. In addition thresholds can also be put on the input 2x2 analogue sums in order to mask out noisy channels but these thresholds are not used directly during the trigger decision making[2].

A L0 trigger signal is sent to the TOR if any of the thresholds are exceeded for any of the sliding window sums. This means that in effect the L0 trigger is just a minimum bias trigger. For the L1 on the other hand the TOR takes into account which threshold is exceeded and which cells are fired. The TOR is discussed further in Section 4.6.2.2.

As indicated in Figure 4.11 the TRU is connected to the same readout and control bus as the FEECs and is therefore also controlled by the same DCS node. It was considered an advantage if the TRU could be treated in a similar way as any other front-end card during the FEE configuration and monitoring and readout. It was therefore decided to implement a *dummy* board controller and a *dummy* ALTRO chip in the firmware of the TRU. From the outside, i.e from the software application interface (API) point of view it *looks like* the TRU has an ALTRO chip and a board controller identical to the FEECs. The word *dummy* reflects that these two devices, the BC and the ALTRO, are not physical devices in the case of the TRU but logical devices inside the firmware, however this is transparent to the software used to control these devices. This has the advantage that a lot of software that was developed for TPC and PHOS can be reused to control and monitor the TRU. An additional advantage is that the RCU sees the TRU as a regular FEEC so that the TRU can be read out like an FEEC without any modification to the RCU firmware, however the content of the data has of course a different meaning for the TRU than for the FEEC and this has to be taken into account during off-line or on-line analysis[3].

4.6.2.2 The Trigger OR

For the full PHOS detector consisting of five modules the Trigger-OR (TOR) receives 40 trigger inputs from the 8x5 TRUs. As already mentioned Chapter 2.4 the L0 latency requirement in ALICE is $1.2\mu s$.

For PHOS this means that in 1.2 μs the particles have to travel from the collision point to PHOS (which takes negligible time), signals have to proceed from the crystals to the FEEC, the FEEC has to send the analogue sums to the TRUs, the TRUs has to issue the L0 trigger to the TOR, the TOR has to send one L0 from PHOS to the CTP, and the CTP has to issue the L0 back to PHOS and eventually to other sub-detectors. This leaves very little time for processing of the trigger information. Therefore, for the

[2]However they are used indirectly since thresholds put on the 2x2 sums will affect the sliding window sums

[3]The TRU can be identified during analysis by the fact that it always has the FEEC address zero

L0 trigger, the TOR, as the name implies, just acts as an OR gate with 40 LO inputs from the TRUs and one L0 output to the CTP.

For the L1 trigger an additional 5.3 μs is available. The TOR should include in the L1 message exactly which TRUs have fired in addition to which thresholds that were exceeded (intermediate, medium or high p_T) by reading out the TRUs. More detailed trigger information will be available during analysis by means of the readout of the dummy ALTROs as described in Section 4.6.2.1.

4.6.3 Flow Control with the *Busy Box*

The control of the data flow between the front-end electronics, DAQ and the trigger is, for PHOS, EMCAL, TPC, and FMD, performed by the so called *Busy Box*. There are physically different Busy Boxes for the different detectors (TPC has two busy boxes).

The Busy Box keeps track of the number of available multi-event buffers in the ALTRO chips[4] and issues a busy signal to the trigger system if the buffers are full.

The Busy Box receives the same trigger as the detectors and is in principle just a counter. The CTP sends a L0/L1 trigger simultaneously to the detectors and to the Busy Boxes. Upon the arrival of the trigger the Busy Box counts one up.

When DAQ has read out the event, the D-RORCs (see Section 2.4.3) sends a signal to the Busy Box which decrements the counter for this equipment. That is, there is one counter for each equipment which is incremented globally upon the arrival of L0/L1 trigger and then decremented one by one for each equipment when the DAQ system has read out the data from this equipment. For PHOS, having 40 equipments (RCUs), this means that for the full PHOS detector (5 modules) there will be one physical connection from the CTP to the busy box and 40 physical connections between the busy box and DAQ.

If the Busy Box detects that the buffers are full it sends a busy signal to the CTP (via the LTU, see Section 2.4.1). It is the CTP which synchronizes the busy signals issued from different detectors (via the Busy Box). If for instance the PHOS Busy Box issues a busy to the CTP then the CTP will block triggers to PHOS, and to detectors participating in a trigger cluster together with PHOS until PHOS releases the busy.

There is no communication between the Busy Box and the electronics. The synchronization between the Busy Box counters and the actual free buffers is done via ECS at the start of run. The detector specific DCS should, as a response to the *START_RUN* command from ECS, then both flush the buffers of the electronics and reset the counters of the Busy Box[5].

[4]So far the Busy Box has only been used by detectors that also uses the ALTRO, however the Busy Box could in principle be used for any other detector.

[5]It has been a problem during the commissioning of ALICE that control of the Busy Box from DCS

DCS node type	Controlled device	API
RCU node	RCU + FEE + TRU	RCU + ALTRO (like) + BC (like)
TOR node	TOR	RCU like
Busy Box node	Busy Box	RCU like

Table 4.3: DCS node flavors and their application interface.

4.6.4 The DCS board

The DCS board [6] was originally designed for TRD but due to its flexibility it was later decided to use it also for TPC, FMD, EMCAL and PHOS. Figure 4.10 shows the DCS board mounted to an RCU. The DCS board is connected to the RCU as a mezzanine card via a general purpose IO bus. The communication between the DCS board and the RCU is provided by the *RCU driver*. The RCU driver was developed primarily in order to provide the communication between the DCS board and the RCU however the RCU driver can be used to control any device that is connected to the general purpose IO bus. Electronics that is controlled by the DCS board over the general purpose IO bus will in the following discussion be denoted by *DCS node*.

4.6.4.1 DCS Node Flavors

The various DCS node flavors are listed in Table 4.3. The fact that the various hardware devices can be abstracted down to three different kind of devices was exploited heavily when writing DCS software. The term *like* as used in this section means that the software application interface is *identical* (Chapter 5).

4.6.5 Connectivity

Figure 4.11 shows one PHOS module from the back side. One module comprises four readout partition, each partition being read out by one RCU. A partition consists of two branches situated vertically on top of each other i.e. branch zero and two read out by RCU 0, branch one and three by RCU 1 and so on. The low voltage for the front-end electronics is provided by two power supplies of type Wiener PL512. The output channels from the LV supplies are connected to two power blocks and then distributed from the power blocks to the eight branches of front-end cards and to the four RCUs as indicated in the figure. The high voltage is provided by one power supply of type ISEG-EHQ 8605p-156-F. The HV supply has eight output channels, one for each branch of front-end

has not yet been implemented for some detectors, including PHOS. The Busy Box then has to be reset manually before the start of a run, something that is often forgotten. If the busy box and the electronics is not synchronized the run typically has to be aborted.

[6] The DCS board is designed and built at the Kirchhoff-institute for Physics, University of Heidelberg. http://www.kip.uni-heidelberg.de/ti/DCS-board/current/

cards. The fact that each FEE branch is equipped with a dedicated HV channel is crucial information in order to understand the noise problems discussed in Chapter 9.

4.7 Readout & Readout Strategy

The raw data is read out via 20 DDLs, one for each RCU, which amounts to four links for each of the 5 modules. The signal from each detection channel is filtered with a semi Gaussian shaper and amplified with two different gains giving 35840 readout channels. The outputs from the shapers are sampled at 10 MHz, giving a time dependent signal of $\sim$10 - 130 samples (the exact number depending on the amplitude of the signal and the configuration of the electronics) of 16 bit[7] words for each readout channel, the amplitude being proportional to the energy deposited in the detection channel and the peak position depending linearly on the time of flight from the interaction point to the detector.

4.7.1 Readout Issues Concerning the ALTRO & RCU

4.7.1.1 ALTRO Configuration Parameters Relevant for PHOS Data Taking

A complete documentation of the ALTRO chip can be found in [44]. Only features relevant for PHOS will be mentioned here. The ALTRO is a mixed ADC and custom integrated circuit signal processing chip designed for the TPC. Upon the arrival of a trigger up to 1008 samples can be processed and stored in a multi event buffer holding either four or eight events[8]. In addition up to 15 pre-samples can be recovered. Each ALTRO chip has 16 independent inputs (ADCs). The behaviour of the chip is controlled with a set of configuration registers. The registers can be divided in two classes, registers global to all 16 channels, and registers dedicated to the individual channels.

As a minimum one always has to specify the number of samples, and typically the number of pre-samples, and to the RCU which channels to read out. In addition the following features of the ALTRO are used for PHOS.

- *Zero suppression:* When the zero suppression is used only samples above a programmable threshold is written to the multi-event buffer of the ALTRO. The zero suppression configuration is global.

- *Data path configuration:*

  Self calibrated pedestal subtraction. The ALTRO subtracts the average DC value the ADC sees outside the sampling window and the actual signal is replaced with this baseline corrected signal. An optional offset can be added in order to recover

[7]The data is sampled with 10 bit resolution but represented as 16 bit words in the PC
[8]With eight buffers only 500 samples can be taken for each event

signals that would otherwise have been less than zero after subtraction. This mode has been regularly used during commissioning of PHOS in ALICE.

Fixed pedestal subtraction: A fixed value is stored in dedicated per channel registers and upon the arrival of trigger this value is subtracted from all the samples of an individual channel. The actual signal is replaced with this baseline corrected signal

Pattern playback mode: A pattern is loaded into the memory of the ALTROs and upon the arrival of a trigger this pattern is played back, replacing the actual signal. i.e the actual signal is discarded altogether. This pattern can be configured per channel, but typically all the channels (of the module) are configured with the same pattern. This mode was extensively used during the commissioning of the second PHOS module (see Chapter 9).

Both the pedestal subtraction mode and the offset are controlled globally whereas the fixed pedestal value is configured for each individual channel. The fact that the zero suppression threshold is controlled globally makes it mandatory to use one of the baseline subtraction modes whenever the zero suppression is used in order to apply the zero suppression efficiently.

The number of samples remaining after zero suppression can be anywhere from zero to the maximum number of samples configured. If the number of samples for an individual channel is zero after zero suppression the payload from this channel will consist of a 40 bit trailer containing the address of the channel and a word specifying that the length of the payload is zero thereby containing no data. For minimum bias p+p collisions, the occupancy in PHOS will be quite low and most of the data written to the multi event buffers of the ALTROs will be these *empty* trailers. Obviously payloads not containing any data are not very useful for analysis and this data can be removed from the readout by the means of *sparse readout* described in the next section.

4.7.1.2 RCU Configuration Parameters Relevant for PHOS Data Taking

There are mainly two kind of configuration parameters relevant for the readout, the active channel list and the *sparse readout*.

Active Channels List The active channel list specifies to the RCU which ALTRO channels are to be read out, and in which order. This order can be important for instance for the clusterizing algorithm in PHOS. The order is provided by writing a list of hardware addresses to a memory in the RCU from DCS during the configuration phase.

Sparse Readout The sparse readout is enabled or disabled by a single bit in one of the RCU configuration registers. upon the arrival of a trigger, the RCU, if the sparse readout

is enabled, asks the BC to set up a list of ALTRO channels containing non-empty data and only channels containing more than zero samples after the zero suppression will then be read out by the RCU.

The setting up of this list takes some time, $\sim$ 50 μs, and the sparse readout should therefore be used only when the occupancy is low. For PHOS it was decided to use sparse readout during p+p collisions, but not for Pb+Pb collisions. where the multiplicity, and thereby the channel occupancy, is higher. Furthermore it only makes sense to use sparse readout when the ALTROs are configured for zero suppression.

Trigger Input Configuration The RCU can receive the L0/L1 trigger input from three different sources. 1) Over an optical link connected to the TTCRx chip on the DCS board (Default) [10]. 2) Via a dedicated NIM input on the RCU[9]. Only a L1 trigger can be issued on this input. 3) By manually issuing a trigger to the RCU from the command line prompt of the DCS board. In this mode the L2 trigger is normally generated by the RCU (see the next section).

Trigger Sequence Configuration The two possible modes are 1) External L0/L1 and external L2. 2) External L0/L1 and internally generated L2 after programmable number of clock cycles. The first option is the only *legal* one, i.e. it is the only one that should be used during data taking in ALICE. The second option is useful for debugging or in situations where one always wants to validate a L0/L1 trigger such as for instance during calibration with electron beam.

4.7.2 Data Format

The smallest entity of data read out from a detector is a *ddl file* containing data from one event and one equipment. A detailed description of the data format is presented in Appendix D. The data contained in the ddl file is formatted on three levels requiring three different steps in the decoding. Firstly the ALTRO creates a back linked list of bunches (sequences of samples above the zero suppression threshold) for each individual channel denoted an *ALTRO payload*. Secondly the RCU creates a back linked list of ALTRO payloads denoted the *RCU payload* from all the channels of that RCU/equipment being read out and wraps the RCU payload in a header and a trailer.

Finally since the ALTRO data bus between the RCU and the FEECs is 40 bit wide the RCU payload is formatted to 32 bit words by the RCU *data formatter*. This formatting is done for four 40 bit words at a time creating five 32 bit words so that the RCU payload can be regarded as consisting of 160 bit words.

[9]This mode is rendered obsolete in the latest version of the RCU firmware

4.7.3 Readout Strategy

The readout of data from PHOS should retrieve as much physics information as possible and at the same time keep the data volume at a minimum. Although it is possible to configure different ALTRO chips with different thresholds it will simplify both the configuration and the analysis to have single global zero suppression threshold. This threshold should be set low (1-2 ADC counts) in order to avoid introducing bias to the data. The self calibrated pedestal subtraction by the ALTRO was tested during the commissioning of the second PHOS module but was found to introduce to much statistical error to the amplitude of the time dependent signal. Firstly, the pedestal values subtracted by the ALTRO is integer values whereas the position of the baseline preferably should be known with floating point precision in order to facilitate the exact reconstruction of the signal amplitudes during the event reconstruction. In addition it was found that for several channels the baseline fluctuated with as much as 4-5 ADC counts after the automatic baseline subtraction. The reason is currently not understood.

The fixed pedestal subtraction mode will be used instead but this mode was not tested so far. The channel dependent baseline (pedestal) value will be measured in dedicated pedestal runs. These baseline values minus an offset will be subtracted from the channels. The offset is added in order to recover signals that would have been less than zero (and thereby ignored by the ALTRO) after subtraction of the baseline. The offset is the same for all the channels so that all the channels, after subtraction, sit on an artificial pedestal signal, defined by the global offset, that is the same for all the channels. The residuals between the floating point pedestal values and the integer values subtracted by the ALTROs will be stored in the Offline Condition Database (OCDB)

With a low zero suppression threshold noisy channels will produce a lot of data. The most noisy channels will be masked out from the readout list. The tool to produce such a mask is presented in Section 9.2.1. The tool was successfully tested during the commissioning of the second PHOS module and it was found that the data volume could be reduced by a factor $\sim$3 by masking out 1 - 2 % of the most noisy channels.

4.8 Results from Beam-tests at CERN PS and Sp$\bar{p}$S

Figure 4.12 shows the energy resolution measured at electron beam tests in the period 2002 to 2006 at CERN PS and SPS. In 2002 and 2003 measurements were done with the PHOS 256 prototype. In 2006 the electron beam test was done with PHOS-1. As seen in the figure the energy resolution satisfies the requirements for $E > 1$ GeV and significantly better for $E < 1$ GeV. The errors are smaller than the size of the symbols and are therefore not shown. It should be noted that for the electron beam test with PHOS-1 the crystals were cooled down to only -17 °C due to problems with the cooling system. The energy resolution is expected to improve if the crystals are cooled down to the nominal -25 °C.

Figure 4.13 show results from measurements of the invariant mass resolution obtained from data take at CERN Sp$\bar{p}$S using a 256 channel PHOS prototype. As seen from the figure an excellent invariant mass resolution was demonstrated for the prototype.

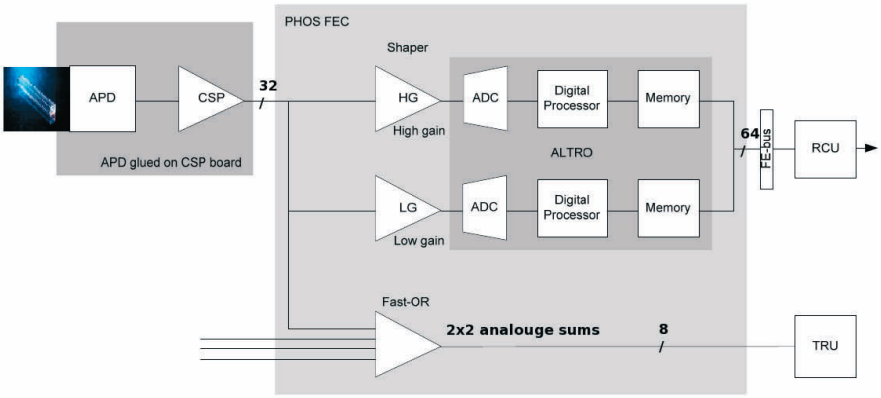


Figure 4.2: Block diagram of the PHOS signal chain including trigger logic. One FEE reads out 32 detection channels (crystal, APD and CSP). The signal is amplified with two different gain giving 64 readout channels. In addition the FEE sends eight 2x2 analogue sums to the TRU (adapted from [10].).

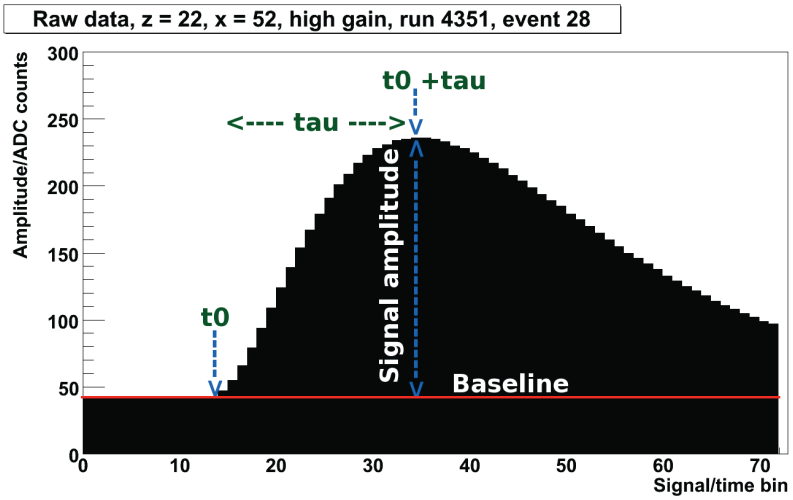
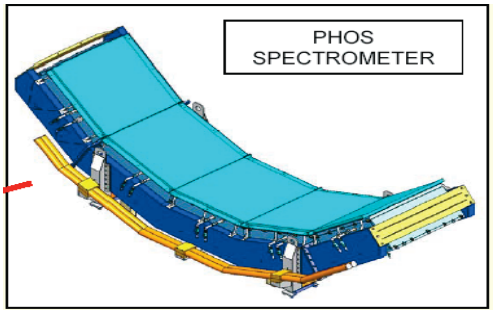
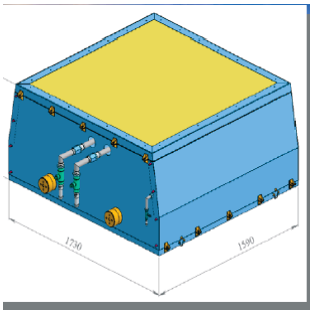


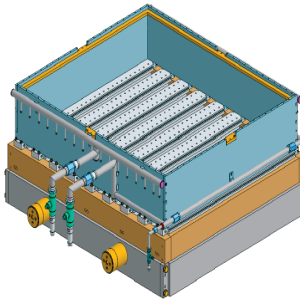
Figure 4.3: Typical digitized signal.



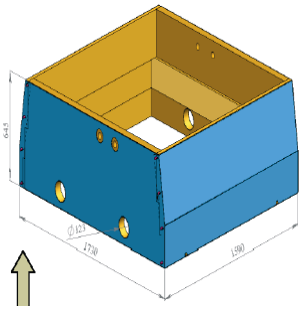
(a) PHOS



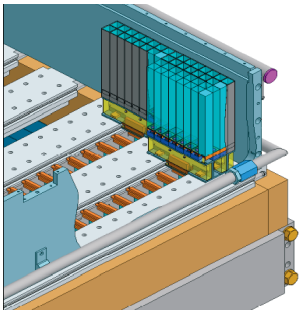
(b) One PHOS module



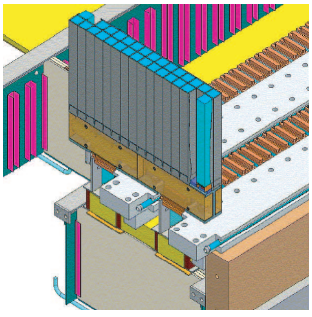
(c) Structure



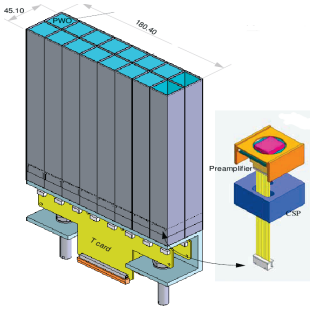
(d) Thermal casing



(e) Mounting

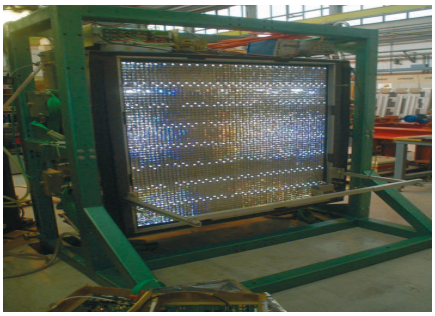


(f) Mounting

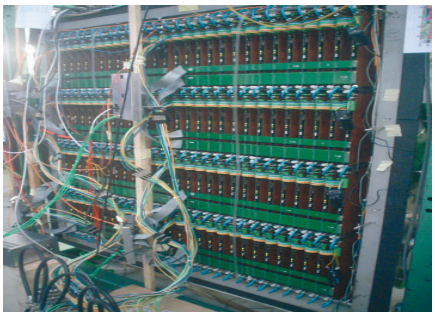


(g) Strip-unit APD and CSP

Figure 4.4: Physical layout.

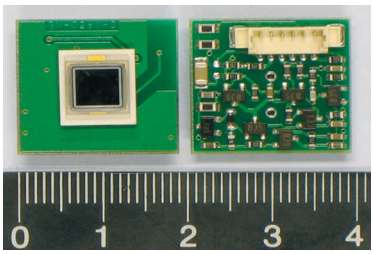


(a) PHOS-2 from front



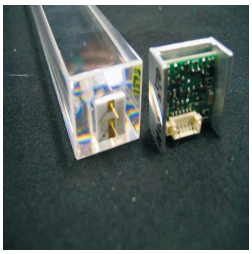
(b) PHOS-2 from back

(c) FEEC, strip unit, APD, CSP, crystal and T-card

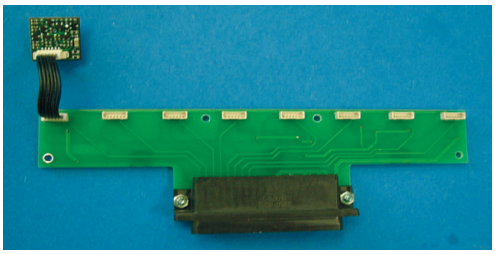


(d) Left: APD, Right: CSP

(e) APD + CSP glued together and the flatband cable for connection to the T-card



(f) Left: The back end of a crystal with the APD glued to it. The two pins sticking out is the connectors for the CSP. Right the CSP



(g) T-Card with one APD+CSP connected

Figure 4.5: PHOS detector components.

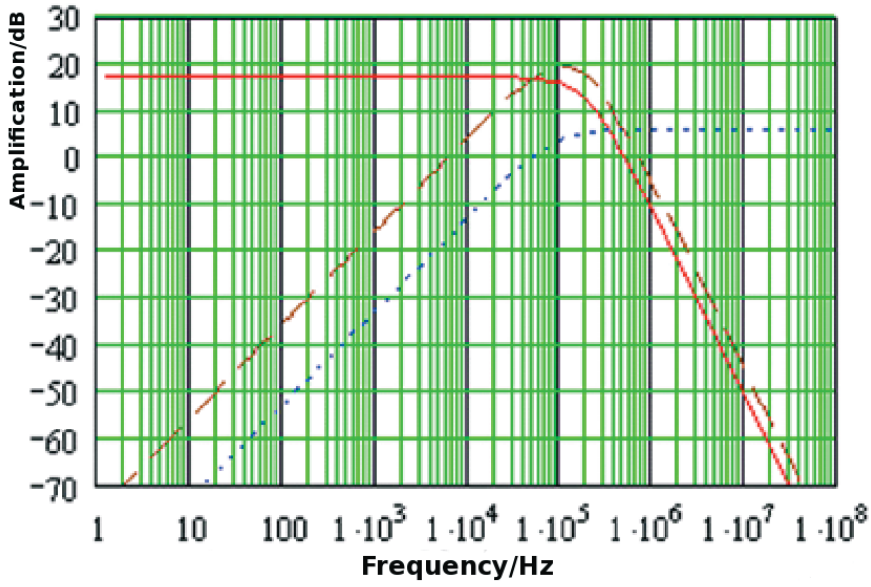


Figure 4.6: The frequency response of the shaper. The dotted blue line is the response of the differentiation stage. The solid red line is the response of the second order integrator and the stitched red line the total response [11].

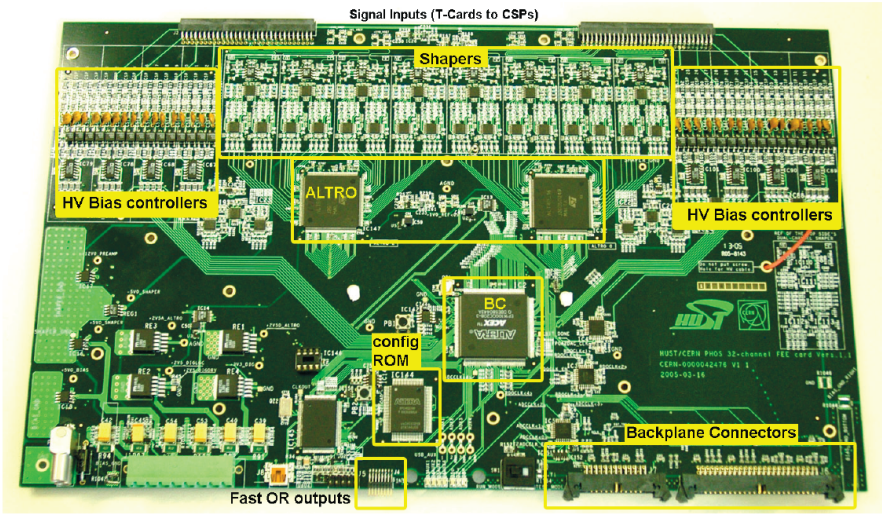


Figure 4.7: The PHOS FEEC. The signals from 2 strip units arrives over the T-Card connectors on top right and top left on the figure. The signal is conditioned with semi Gaussian *shapers* and digitized with four ALTRO chips (two on each side of the FEEC). The digitized data is subsequently read out over the GTL back-plane connected to the *back-plane connectors*. The readout is controlled by the Board Controller (BC). Per channel APD HV bias voltage is also controlled by the BC. The user defined individual per channel HV bias voltage is loaded into the BC. The BC applies these setting to the *HV Bias controller* which comprises 8 DACs with 4 outputs each for the 32 detection channels. In addition the BC continuously monitors the *health* of the FEEC: mainly temperatures, voltages and currents. Fast analog signal outputs are provided by the *fast OR* outputs which are connected to the TRU. Each of the fast OR outputs is an analog sum of 2x2 detection channels so that there is a total of eight analog outputs per FEEC [10].

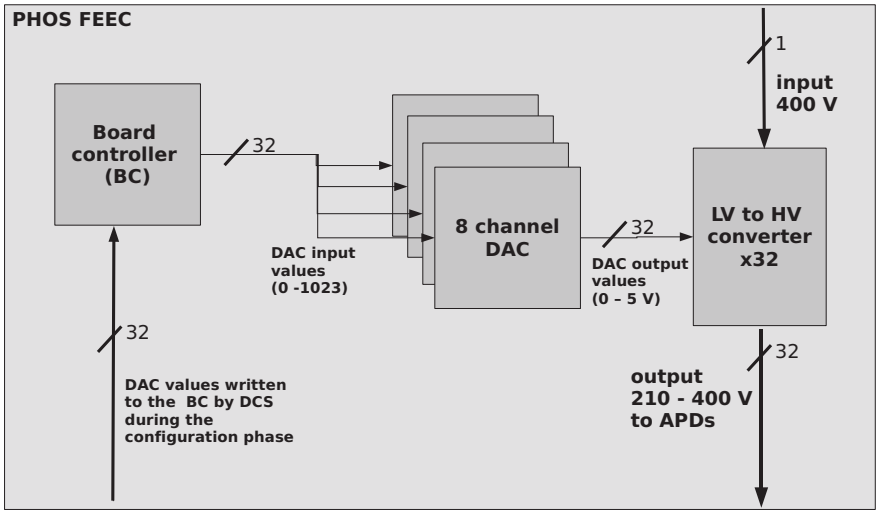


Figure 4.8: Schematic picture of the hardware view of the APD bias control.

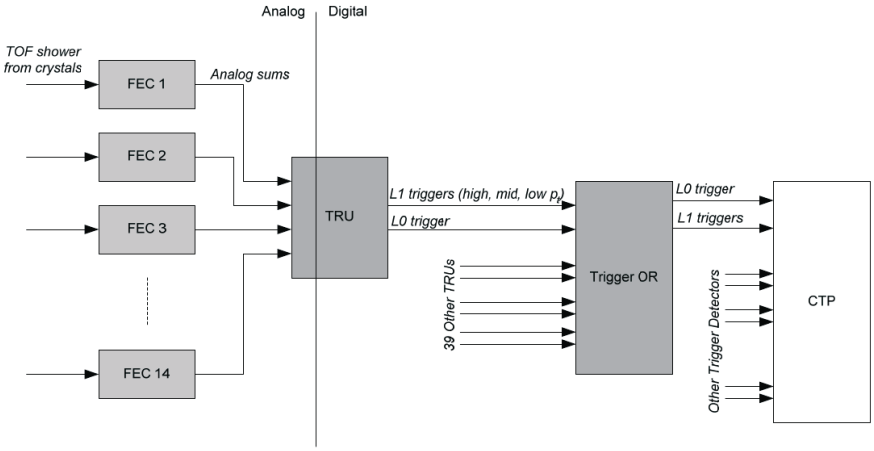


Figure 4.9: Sketch emphasizing one trigger region of PHOS to show the data flow of the trigger detector functionality [10]. There is one TRU per branch of FEECs, eight per PHOS module so that there is a total of 40 trigger inputs from PHOS to the TOR.

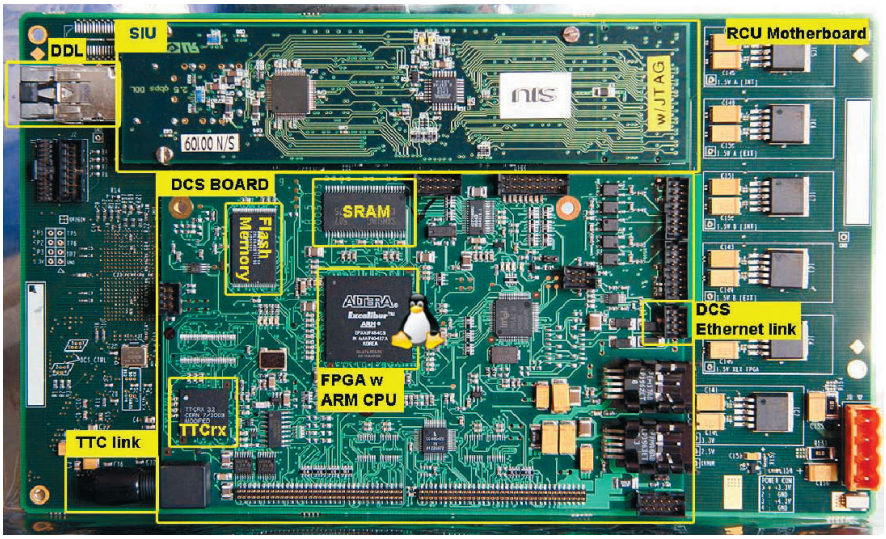


Figure 4.10: RCU with the DCS board and the SIU mounted [10].

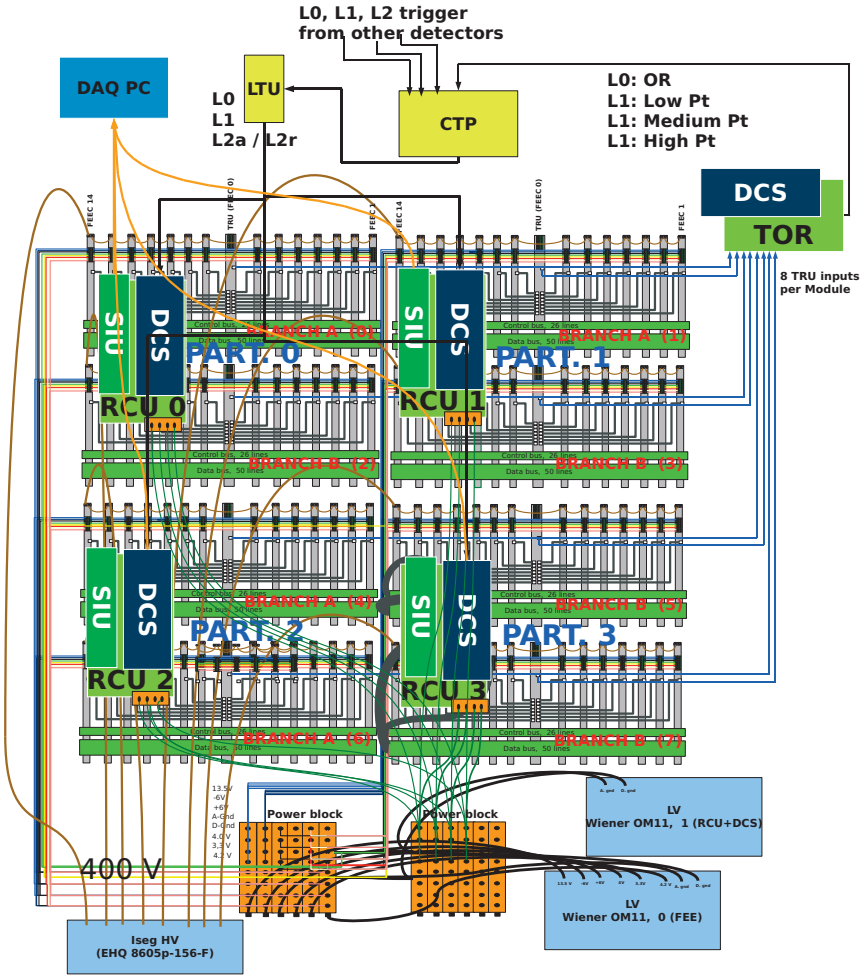


Figure 4.11: One PHOS module as seen from back.

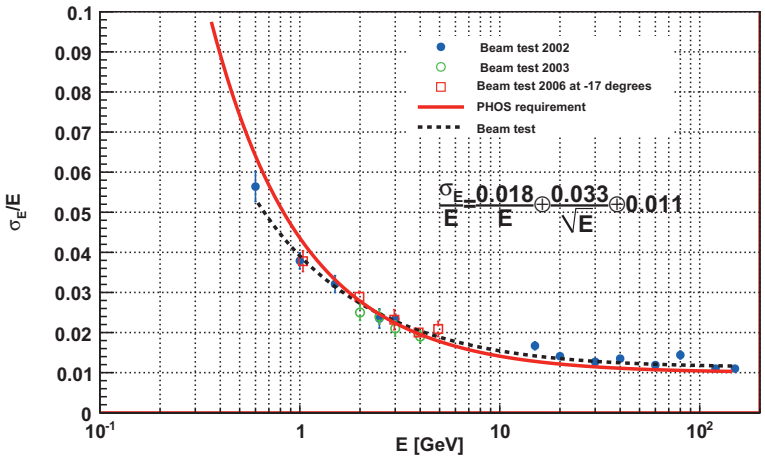


Figure 4.12: Energy resolution measured for all beam tests [11].

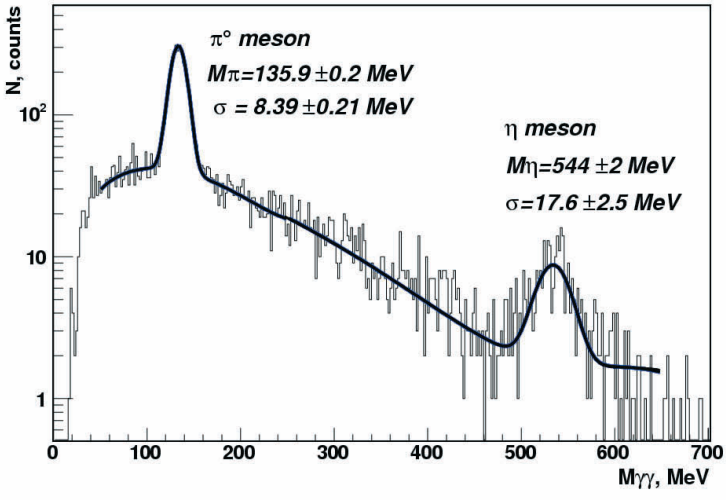


Figure 4.13: Invariant mass resolution measured at CERN SpP̄S [12].

Chapter 5

The PHOS Detector Control System (DCS) for Front-End Electronics (FEE)

5.1 Introduction

The ALICE DCS system was introduced briefly in Chapter 2.4.2. This chapter will focus on the detector control system for the PHOS FEE.

The FEE DCS system for PHOS has to perform the following tasks.

- Configuration

  Configuration of HV of the APDs

  Configuration of the ALTRO chips

  Configuration of the RCU

- Monitoring

  Monitoring of temperatures, currents and voltages of the FEECs

The DCS system can be divided logically into three layers: The *supervisory layer*, the *control layer* and the *field layer* as indicated in Figure 5.1. Referring to the figure a light blue color indicates components common to all sub-detectors. The framework chosen for the supervisory layer for all the four LHC experiments is Prozess Visualisierungs - und Steuerungs-System (PVSS).

Yellow denotes software components common to TPC, PHOS, CPV, EMCAL, FMD and TRD which are identical for these detectors in the sense that software is compiled from the same source code, however they are typically configured differently for different detectors.

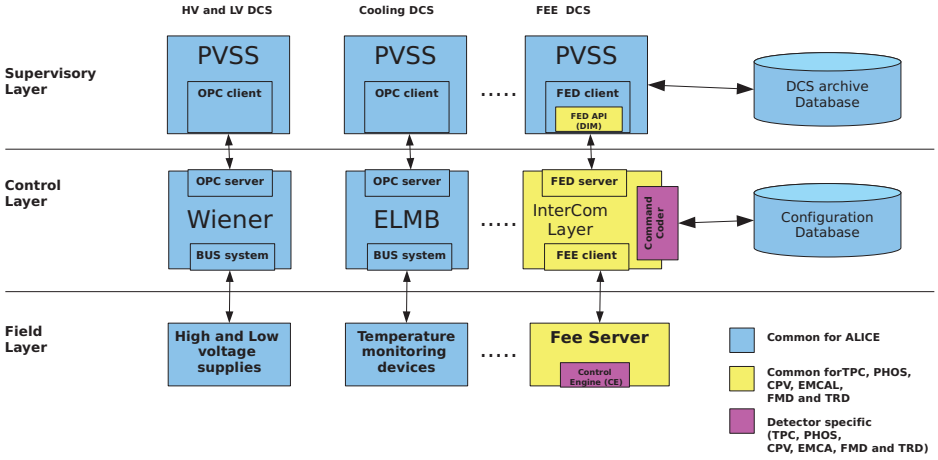


Figure 5.1: Logical overview of the Detector Control System ([10]).

It was considered a design requirement that the detectors listed above use software that is compiled from identical, not just similar, source code thereby facilitating software reuse and maintenance by keeping the duplication of software to a minimum. The *yellow* components comprise the *Front End Device (FED) API, FED server, InterCom Layer (ICL), fee-client* and the *FEE server* (explained below).

The *Command Coder (CoCo)* and the Control Engine (CE), color coded with purple, are the only detector specific components.

Figure 5.2 shows the various field layers for PHOS and the communication between them. The division of the layers is valid for all DCS sub-systems (for all detectors), including the DCS for the front-end-electronics.

When communicating with the RCU the user application communicates directly with the RCU main registers via the kernel RCU device driver. One such application that was extensively used during commissioning of PHOS, is the RCU shell *rcu-sh* [47].

5.2 Interfaces

The communication with peripheral devices, i.e. the ALTRO chips and the board controllers, is provided by two interfaces contained in the RCU main FPGA, the *ALTRO bus interface* (AIF) and the *slow control (SC) interface* (SCIF). The *ALTRO bus interface* provides two-way communication between the RCU and the ALTROs and between the RCU and the board controllers over the ALTRO readout bus. The SCIF provides a

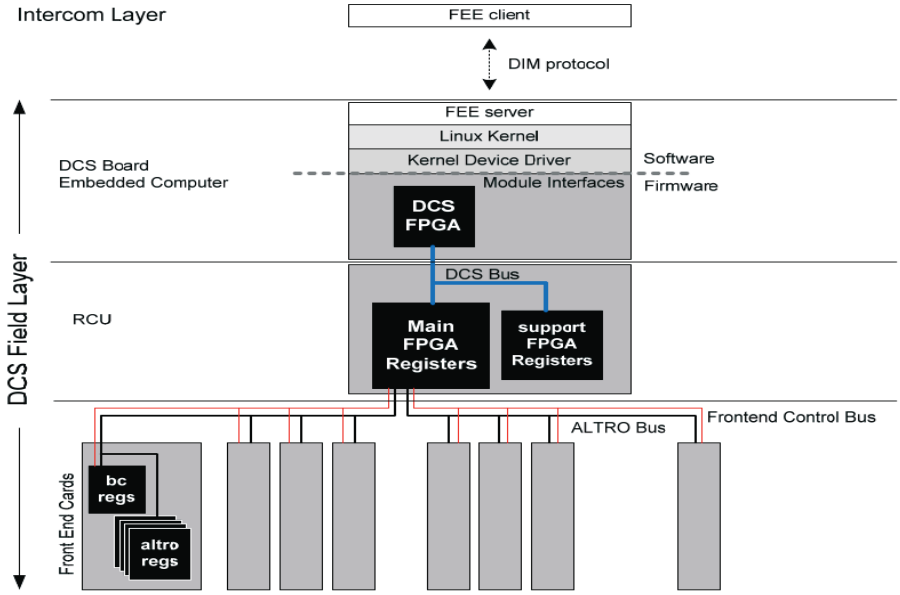


Figure 5.2: The field layer devices for PHOS. The SC communicates over the Front-end Control (FC) bus [10].

two-way communication between the RCU and the board controller over a dedicated I^2C connection, however, this interface cannot access the ALTROs.

The task of the SC is to provide monitoring of temperatures, voltages and currents which are read from the BC. In addition the SC should handle interrupts from the individual FEECs (5.4). Further documentation on the configuration and control of the PHOS board controller, the ALTRO chip and the RCU can be found in [10], [45] and [48] respectively. Referring to Table 5.1 it is useful to note for the following discussion that the ALTRO can only be accessed over the ALTRO bus so that during the configuration phase the AIF has to be used for the configuration of the ALTROs. In addition, as the name implies, the *slow control* is slow, at least compared to the AIF so that even if it is possible to configure the board controller over the slow control it is preferable in terms of performance to configure also the board controller over the ALTRO bus. The FEE configuration software for PHOS therefore relies entirely on the use of the ALTRO bus.

During normal operation the detector is controlled from the supervisory layer indirectly via ECS (Section 2.4.1). It is also possible to interact directly from PVSS or with the fee client.

ECS assumes a given fixed set of run types (physics, cosmic, pedestal run.. etc...) and

each run type has typically, depending on the detector, a given detector configuration associated with it. For instance, using PHOS as an example, one would like to configure the electronics with hardware zero suppression for physics or cosmic runs but not for pedestal runs.

5.3 Configuration

During the configuration phase, run specific configuration parameters are applied to the electronics. In general the configuration parameters depend mainly on the run-type received from ECS at start of run. In addition, during commissioning and debugging several different sets of run parameters for the same run type will be tried out to find the configuration parameters that best satisfy the physics requirements.

The supervisory layer receives information about the run type from ECS and translates it to a unique configuration *key*. If PVSS is controlled directly the user selects a configuration from a drop down menu on the top level user interface and the selected configuration is translated to a unique key. The key is sent to the InterCom layer which communicates with the supervisory layer over the FED API and the FED server. The ICL fetches the corresponding configuration from the *configuration database*. In this database the data is stored in human readable form. The command coder translates the configuration data to a form that can be understood by the electronics i.e it has the equivalent functionality as that of a compiler.

The ICL adds a header to the coded block marking it as data that should be used to configure the electronics. The coded configuration data together with the header is send by the ICL to the DCS board using the fee-client and the fee-server. The fee-server reads the header of the block identifying that it should be sent to the control engine for execution. The actual configuration of the electronics is done by the control engine using the interfaces described in Section 5.2.

5.3.1 Configuration Parameters

The parameters and devices that need to be configured during the configuration phase are listed in Table 5.1. Each entry in the table is described in more detail in Section 4.6.1.

5.4 Monitoring

Monitoring of temperatures, voltages and currents are important for a safe operation of the PHOS detector. For PHOS, as well as for EMCAL and TPC, temperature, current and voltage sensors are placed on different locations on the FEEC. The sensor values are

Parameter	Device	Access
Trigger path	RCU	RCU driver
Active channel list	RCU	RCU driver
Active FEE list	RCU	RCU driver
APD DAC settings	BC	AIF or SCIF via the RCU driver
Number of samples per readout ch.	ALTRO	AIF via the RCU driver
Baseline correction mode	ALTRO	AIF via the RCU driver
Zero suppression mode	ALTRO	AIF via the RCU driver
Thr. for the HW interlocks	BC	AIF or SCIF via the RCU driver

Table 5.1: Left: PHOS configuration parameters. Middle: devices where these parameters are configured. Right: available access modes.

sampled with dedicated ADCs and written to dedicated register addresses of the BC. The complete documentation of the BC for PHOS can be found in [10].

The CE reads temperatures, voltages and currents continuously whenever the electronics (FEECs mainly) is powered on. In addition certain error and status registers are monitored. The CE publishes the monitoring values as DIM services using the fee-server/client channel. The ICL subscribes to these services and forward the monitoring values to PVSS using the FED API/FED server connection.

The kind of services available is configurable in all three layers. In the field layer the CE is configured for which sensors situated on the FEECs to read out. In addition configurable dead bands are applied and monitoring values are only updated and published if the changes in individual monitored values are larger than the dead band.

In the control layer the configuration of the ICL determines which services the ICL should subscribe to and which monitoring values to forward to the supervisory layer. This is specified in a configuration ASCII file containing a list of services. In addition it is the task of the ICL to send data to the supervisory layer at reasonable intervals. It can happen for instance that a lot of temperature values are updated very frequently when for instance powering on the electronics (because the temperature will increase continuously before it stabilizes) and it is the job of the ICL to suppress the update frequency in order to not flood the supervisory layer with too much data.

On the supervisory level the monitored values are connected in PVSS to so called *data points*. These data points can be monitored and the values archived. Typically only a subset of the data points are archived. A value that is archived ends up in the *DCS archive* database. Data that is archived is automatically available during off-line analysis.

5.5 Hardware interlocks

If any monitoring values are out of range the FEEC issues an interrupt[1] to the slow control module of RCU and it is the SC which ultimately, depending on how it is configured, handles the interrupt. The types of errors/interrupts can be classified in *hard* and *soft* where *hard* means an error that can cause physical damage to the FEEC, for instance a temperature above a given limit, and a *soft* is an error that can cause malfunction of the FEEC but that will normally not cause any physical damage (for instance digital voltage below threshold).

The monitoring has to run whenever the PHOS electronics is powered on, including of course during data taking. The latter has the implication that the ALTRO bus should not be used for monitoring since during data taking the ALTRO bus is used for reading out data from PHOS. Even though it is technically possible, it could lead to fatal interference with the data readout and in the worst case the abort of a run.

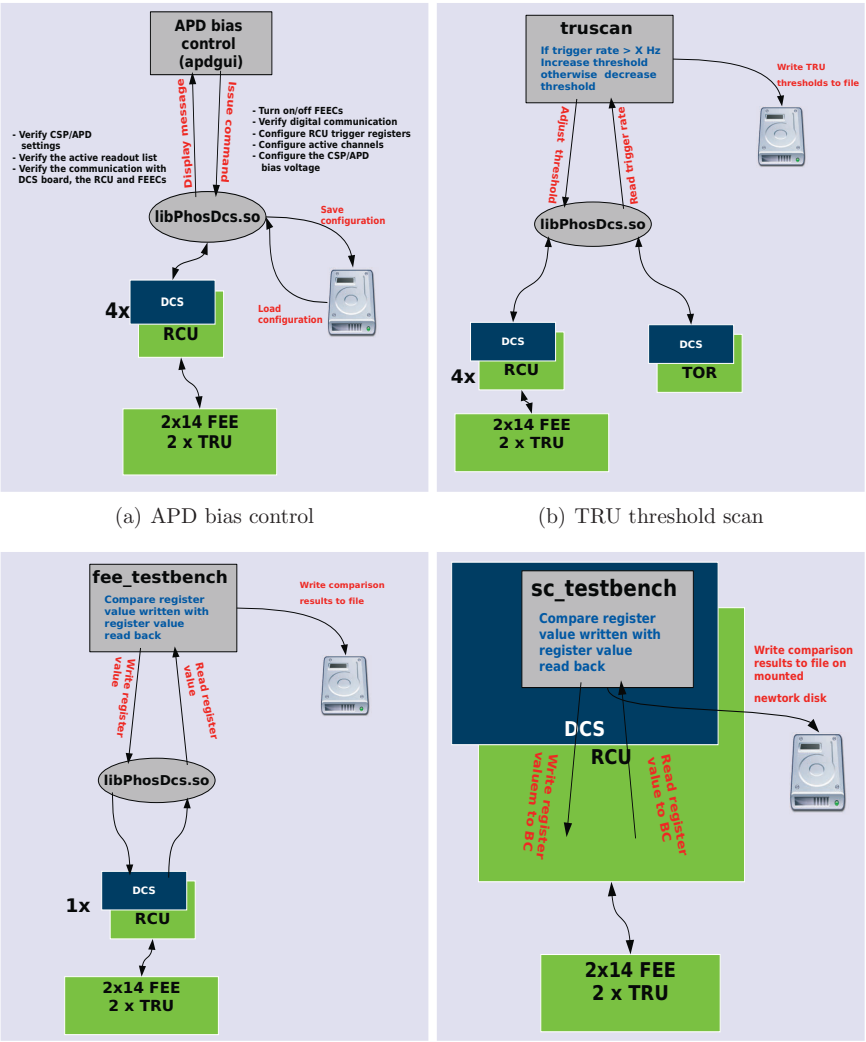
5.6 Software Organization and Design

The DCS control software for the FEE is used not only for the normal running and experimental data taking, but also for debugging and commissioning of the PHOS detector. A single FEE library was developed in order to satisfy both the need for debugging and commissioning in the lab as well as for the normal running of the detector. The design philosophy was to have a single FEE control library accessed in different ways by different DCS tools. For regular physics data taking only a limited set of functionality is needed, however for debugging purposes an extensive set of functionality is available in this library.

Several tools to control and debug PHOS and the PHOS FEE were developed and successfully used during the commissioning of PHOS and during installation in ALICE. The key reason for the success was the virtualization of the driver for the PHOS front-end-electronics described in section 5.6.1. The most important ones and their usage are listed in Table 5.2. The APD bias control software is described in detail in Appendix J. The four programs listed in Table 5.2 are represented schematically as four use cases in Figure 5.3.

Apdgui The PHOS DCS library *libPhosDcs.so* was primarily designed for the *apdgui*. As the name is implies the primary mission of the apdgui was to provide a graphical user interface for the control of the HV of the APDs. However over the last 2 - 3 years many more features were added and the apdgui now does much more than controlling the HV of the APDs (refer to Appendix J). Nevertheless the name apdgui was kept for historical reasons.

[1] The interrupt can be disabled by the configuration of the BC



(a) APD bias control

(b) TRU threshold scan

(c) FEE testbench

(d) Slow control testbench

Figure 5.3: Four different use cases of the PHOS DCS software

Program name	Functionality	Communication
apdgui	APD bias and readout control	ALTRO bus
truscan	Adjustment of the trigger thresholds for the TRU	ALTRO bus
feetestbench	Low level transaction test of the FEE	ALTRO bus
sc_testbench	Low level transaction test between the DCS board and the board controller	Slow Control (SC)

Table 5.2: DCS software applications for PHOS.

The new features includes the configuration of the ALTROs (number of sample and pre-samples). The most recent version of the gui also controls the baseline subtraction and zero suppression modes of the ALTRO, but these new features were not yet extensively tested since most of the data so far was taken without baseline subtraction or zero suppression.

The subset of *libPhosDcs.so* providing access and control of the APD HV configuration will be denoted in the following as the PHOS APD Bias Control (PABC). The apdgui provides a graphical user interface to all the functionality of *libPhosDcs.so* including the PABC and was successfully used during all phases of commissioning of PHOS starting with the first PHOS module in 2006. The main task of the *apdgui* is, as the name implies, to control the HV APD bias settings of the PHOS detection channels. The apdgui also controls the on/off powering of the individual FEECs as well as the configuring of the readout settings i.e. the active channel list of the RCU and the ALTRO configuration (number of samples, pre-samples etc..). As indicated in Figure 5.4 and 5.3(a) the apdgui can also be used as an FEE configuration editor. The user can modify the FEEC configuration in the graphical interface to the APD bias control software (contained in libPhosDcs.so) and save a new configuration by clicking on buttons in the apdgui. By default the most recently saved configuration will be loaded upon the next execution of the apdgui. Further documentation of the PHOS APD bias control software can be found in Appendix J.

TRU Trigger Scanning The purpose of the TRU trigger scanning is to adjust the triggering threshold of the TRU. It is foreseen that the TRU should trigger on three different levels listed below.

- Minimum bias
- Intermediate p_T
- High p_T

Since the main focus on the use of the TRU so far was cosmic calibration, only the minimum bias mode will be considered in this thesis. The use case scenario for TRU

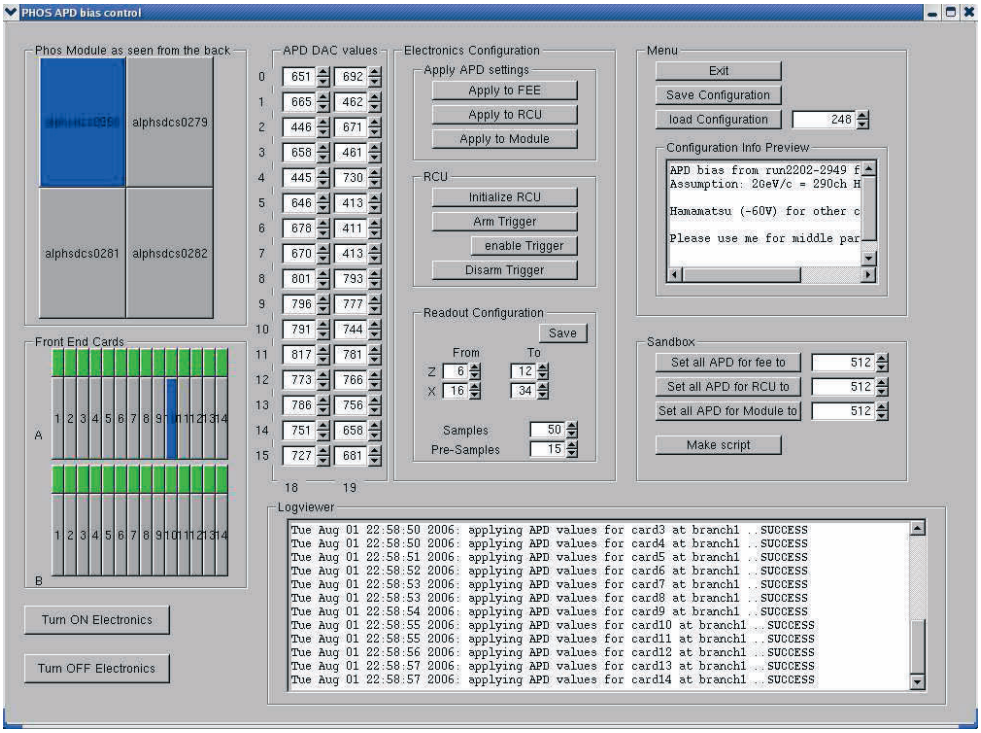


Figure 5.4: The PHOS APD Bias control (PABC/apdgui) Graphical User Interface (GUI)

scanning is given in Figure 5.3(b). For each 2x2 analogue sum all except a given 2x2 analogue sum is blocked, i.e. the triggering threshold is set to the maximum possible value for the 2x2 sums blocked and the trigger threshold is increased/decreased for one 2x2 sum at the time. The threshold is found by a simple binary search as shown in Figure 5.3.

- If the trigger count read from the TOR is *zero* over a 3 second interval the triggering threshold is decreased
- If the trigger count read from the TOR is higher than *zero* over a 3 second interval the threshold is increased

Fee_test_bench The FEE fee_test_bench suite was developed in order to test the basic communication with the FEE and was used extensively during the commissioning of the

first and second PHOS module. The testbench incorporates simple transaction tests where arbitrary values are written to RCU, ALTRO, or BC registers and the value written is then subsequently read back for comparison. For the BC, the registers can be accessed either via the ALTRO bus or over the I^2C serial line.

Sc_test_bench The sc_test_bench software was developed in order to test the communication between the RCU and the BC over the I^2C serial line (slow control line). It performs the same task as fee_test_bench when communicating with the BC except that the digital communication is done over the I^2C serial line (i.e. over the FC bus) instead of over the ALTRO bus.

Unlike the other tools the sc_test_bench has to be run directly on the DCS board. It would clearly have been an advantage if the sc_test_bench was an integrated part of the PHOS DCS library. The reason why this is not the case is that at the time the DCS library was developed the I^2C connection did not work for PHOS so that there would have been no way to test and debug the software.

5.6.1 The Virtual PHOS FEE Driver

The access and control of the PHOS FEE can be grossly simplified by noting that there are mainly two operations involved; writing and reading registers. Whenever registers are written one normally wants to read the values back from the electronics for verification, however read-only operations should also be possible.

5.6.1.1 The Low Level Interface

The two basic building blocks are therefore the two functions *WriteReadRegister* and *ReadRegister* implemented in the *PhosFeeClient* class in the PHOS DCS library. Their interfaces are shown below.

```
class PhosFeeClient :public FeeSampleClient, public PhosDcsBase
{
 public:
  PhosFeeClient();
  virtual ~PhosFeeClient();
  const int WriteReadRegisters( const int devType, const char *hostname,
                                const unsigned long *regs,
                const unsigned long *values, const bool *verify,
        const int N, const int branch, const int feenr,
                                const int altro, const int channel);


  int ReadRegisters( const int devType, const char *hostname,
                     const unsigned long *regs, unsigned long *rbValues,
                     const int N, const int branch, const int feenr,
                     const int altro, const int channel);
```

```
   ..........
   .........

}
```

The user specifies what type of device to access with the parameter *devType*. The *devType* can be either RCU, BC or ALTRO.

The equipment to control is given by the *hostname* of the DCS board controlling the equipment. If the apdgui is used the mapping between geometrical position of the equipment and the hostname of the DCS board controlling it is done automatically transparently to the user.

The parameters branch, feenr, ALTRO and (ALTRO) channel have default values and does not have to be specified if the RCU is accessed (devType = RCU). If the BC is accessed then the parameter *branch* and *feenr* must be set. If individual ALTRO channels is accessed it is necessary to specify in addition the parameters *ALTRO* and *channel*.

For the *WriteReadRegisters* function register addresses and values to write to these addresses are given by the arrays **regs* and **values*. The length of the arrays are specified by *N*. The array **verify* specified which registers should be read back for verification.

If the values read back are the same as those written (**values*) the function returns zero. Otherwise it returns an error code indicating what went wrong in the transaction. If the *devType* is set to ALTRO but *ALTRO* and *channel* is not specified then the ALTROs are configured in broadcast mode. The values are not read back for verification in this case.

For the *ReadRegisters* function the parameter **regs* specifies an array of registers to read. The values of these registers are then returned in the array **rbvalues*.

5.6.1.2 The High Level Interface

The basic building blocks described in Section 5.6.1.2 are used to build higher level commands. These higher level commands are collected in the interface class *DcsInterface* of the DCS library. All DCS applications communicate with the electronics via this interface. A part of this interface class is shown below.

```
class  DcsInterface : public PhosDcsBase
{
 public:
                  DcsInterface();
  virtual         ~DcsInterface();
  void            ApplyApdSettings( const int modID, const int rcuId, const int branch,
                                    const int card, char *messageBuffer) const;
  void            ApplyTruSettings( int modID, int RcuID, char *Mesbuf, unsigned long *regAddr,
                                    unsigned long *regVal, bool *verify, int nTruRegs);
  int             ArmTrigger( const int modID) const;
  unsigned int    CheckFeeState( const int mod,  const int rcu , const int branch ,
```

```
                                const int cardId, char *tmpMessage);
  void            DisArmTrigger( const int modID, const int RcuID, char *messageBuffer) const;
  void            EnableTrigger(int modID);
  void            EnableTrigger_ttcrx(int modID);
}
```

Chapter 6

Extraction of PHOS detector signal parameters needed for physics analysis

6.1 Reconstruction of PHOS data - General Overview

The processing of data from any detector, including PHOS, is in general classified as either event reconstruction or analysis. This chapter deals primarily with the event reconstruction.

The goal of the event reconstruction is to describe events as accurately as possible in terms of Particle Identification (PID) and kinematic variables.

The reconstruction is done first for individual sub-detectors. The event fragments from individual detectors are then subsequently merged into what in ALICE is called Event Summary Data (ESD) describing global event characteristics. The ESD is a C++ class and the reconstruction creates one instance/object of this class for each event in a given run.

The data is read out from any detector, including PHOS, on a format that is in general non-readable for the computers that have to process it. The first step is to *decode* the data in order to make it readable for a PC. In the case of PHOS, EMCAL and TPC this is done as described in Appendix D.

Furthermore the individual readout channel has to be assigned to, or *mapped*, to a geometrical location relative to the ALICE coordinate system. The mapping involves three steps: Firstly, the mapping from the equipment ID (refer to Section 2.4.3) to the detector readout partition. For instance for PHOS this mapping should determine which one of the five PHOS modules the equipment belongs to and at which location within the module the equipment is located. The next step involves mapping from the hardware address for individual readout channels to their geometrical position relative to the equipment. For

PHOS this step implies translation of the hardware address given as FEEC, ALTRO chip, ALTRO channel and branch (refer to Section 4.6) to *(z,x)* crystal coordinates.

Finally the real world geometrical coordinates are determined by survey measurements. The survey is performed by for instance measuring the actual positions of crystals inside the PHOS module relative to a reference point for the module. The discrepancy between the nominal position of the detector elements and the measured ones is called *misalignment*. An accurate determination of the misalignment is necessary in order to achieve precise event reconstruction, crucial for the physics performance.

The information about the mapping and the misalignment is stored in the Offline Condition Database (OCDB) that is used during off-line reconstruction and on-line reconstruction in the HLT. The OCDB has of course to contain the correct information about the mapping[1].

Once the data has been decoded and the correct mapping applied, the reconstruction is performed in the following steps described in the subsequent sections: Extraction of amplitude and time of flight for individual readout channels for instance by using the Peak-Finder algorithm (Section 6.2.1), conversion from ADC amplitude to real energy, clusterizing, evaluation of geometrical location of the hit and the time of flight from the collision point.

Figure 6.1 shows a typical event from data taken with 2 GeV electron beam at CERN PS in 2006. Figure 4.3 shows the channel in the middle. Each of the 9 plots in the Figure 6.1 corresponds to one crystal with the plots arranged in the same order as the crystals. Only data for the high gain readout channels is shown. The Figure shows the time dependent signals read out when a single electron hitting the middle crystal creates an electromagnetic shower which propagates into neighbouring channels. Referring to Figure 4.3, the energy deposited in a single channel is proportional to the amplitude after subtraction of the baseline. The peak position indicated by $t_0 + \tau$ depends linearly on the time of flight from the collision point to PHOS. For slower particles the peak position will be located further to the right. The first step in the reconstruction is to evaluate the amplitude and the time-of-flight from a time dependent signal, the *samples*, sampled by the ALTRO ADC as indicated in Figure 4.3.

Two methods, or *estimators*, are discussed in Section 6.2.1 and G.1: The LMS and the Peak-Finder . There is an additional estimator that was extensively used, denoted the *crude* estimator which takes as the amplitude the baseline-subtracted maximum sample value, and as a time-of-flight estimate the bin of the max sample value. As the name implies this estimator is not very accurate. However it is the most robust one. In addition it is model independent. When dealing with for instance a very noisy signal, as described in Chapter 9, the crude estimator was the one mostly used.

Once the amplitude and peak position are known for individual channels the next

[1]During the commissioning of the second PHOS module it was found that for three out of four DDLs the optical fibers was actually not connected according to the database

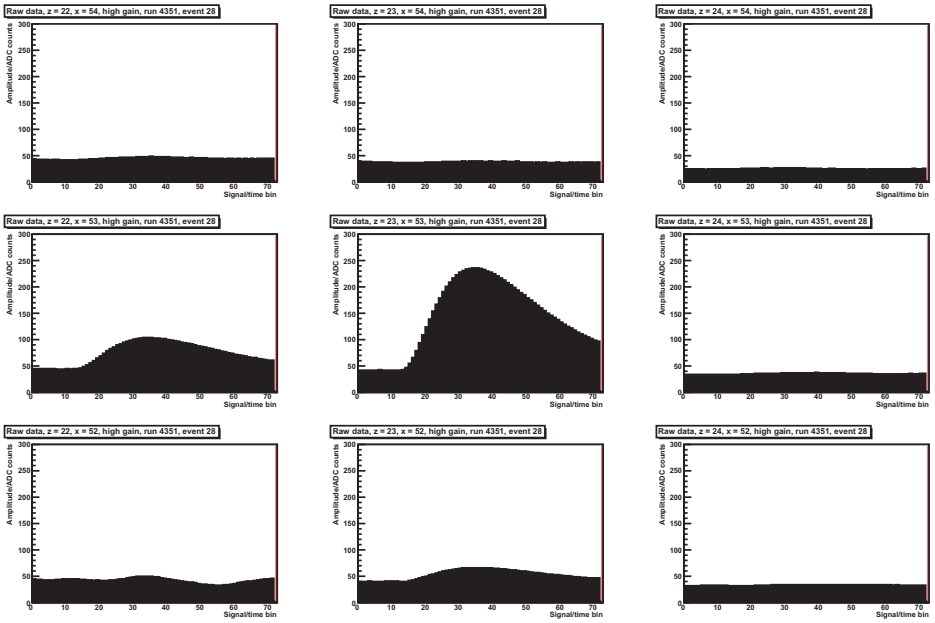


Figure 6.1: Event from electron beam in 2006

step is to use this information to form clusters. This process is performed during the *clusterization*. A cluster is defined as a collection of adjacent cells fired by a single particle. The aim of the Particel Identification (PID) is for PHOS to determine whether or not this particle is a photon.

6.2 Extraction of Amplitude and Peak Position for Individual Readout Channels

The energy and time-of-flight are the only two parameters to be extracted for individual readout channels. The signal shape is assumed to be constant, i.e approximately a gamma-2 function (as given by Equation 4.13) for an individual channel from event to event. It is not foreseen that any event by event pulse shape analysis will be needed.

The most commonly used technique is to perform a least square fit to an anticipated model. In the general case one uses the Least *Mean* Square (LMS) method, where *mean* emphasises that the first and second order statistical properties, i.e. the covariance matrix is used (see Chapter G.1). If the full covariance matrix is unknown, one typically uses a *chi-square* fit which represents the special case of the LMS algorithm when the covariance matrix is a diagonal matrix, in the simplest case the identity matrix. The LMS algorithm is discussed in more details in Section G.1.4.

We shall call a method or algorithm that evaluate the energy and peak position for an *estimator*. The desirable properties of an estimator is discussed in detail in Section G.1.3.

One of the most important features of an estimator is the variance of the parameter estimates. If the variance is as small as possible then the estimator is said to be *efficient* or some times also *fully-efficient*. The LMS is an *efficient* estimator.

Another important criteria, especially for the HLT, is performance in terms of speed. LMS algorithms are typically relatively slow, particularly for non-linear models.

Another approach, the Peak-Finder algorithm [49] and [50], satisfying the performance requirements for the HLT, has been implemented and tested. Since the LMS is a commonly used and fully efficient estimator it is natural to compare the Peak-Finder with the LMS in terms of efficiency and performance. The result of this comparison is presented in Appendix G. To summarize the results, the Peak-Finder is also fully efficient. Furthermore it ∼three orders of magnitude faster, the exact number depending on factors such as for instance the number of samples (see Appendix G).

6.2.1 The Peak-Finder Algorithm

This section gives a brief introduction to the Peak-Finder algorithm. More details can be found in Appendix G.

The fastest computing time is achieved for estimators that are linear function of the measurements. The Peak Finder (PF) algorithm aims to evaluate the amplitude and peak

position as a weighted sum of the samples $\boldsymbol{m} = [m_0, m_1, \cdots, m_{N-1}]$ using statistically optimized weights. The work presented in this Section is based on [49] and [50]. For the evaluation of energy the objective is to find a set of weights $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_{N-1}]$ so that

$$\tilde{A_{adc}} = \boldsymbol{m} \cdot \boldsymbol{\alpha}. \tag{6.1}$$

This constraint seems reasonable at first glance since $E\{\boldsymbol{m}\}$ scales linearly with the amplitude of the signal.

The TOF, hoverer, obviously does not scale linearly with the amplitude. For the evaluation TOF we shall therefore aim to obtain a set of weights $\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_{N-1}]$ that gives an unbiased estimate of Δ_{t_0} for a given amplitude. This given amplitude might as well be chosen to be one. Δ_{t_0} is then estimated by normalizing $\boldsymbol{m}$ to unit amplitude. That is

$$\tilde{\Delta_{t_0}} = \frac{\boldsymbol{m} \cdot \boldsymbol{\beta}}{\boldsymbol{m} \cdot \boldsymbol{\alpha}} \tag{6.2}$$

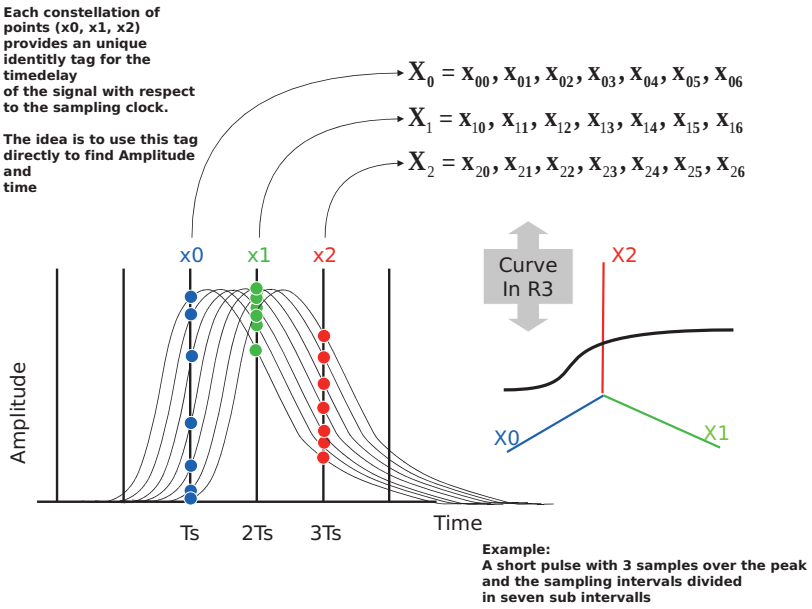


Figure 6.2: The representative curve assuming three samples and seven subdivisions of T_s. The solution to the Peak-Finder equations is equivalent with fitting the curve with a plane.

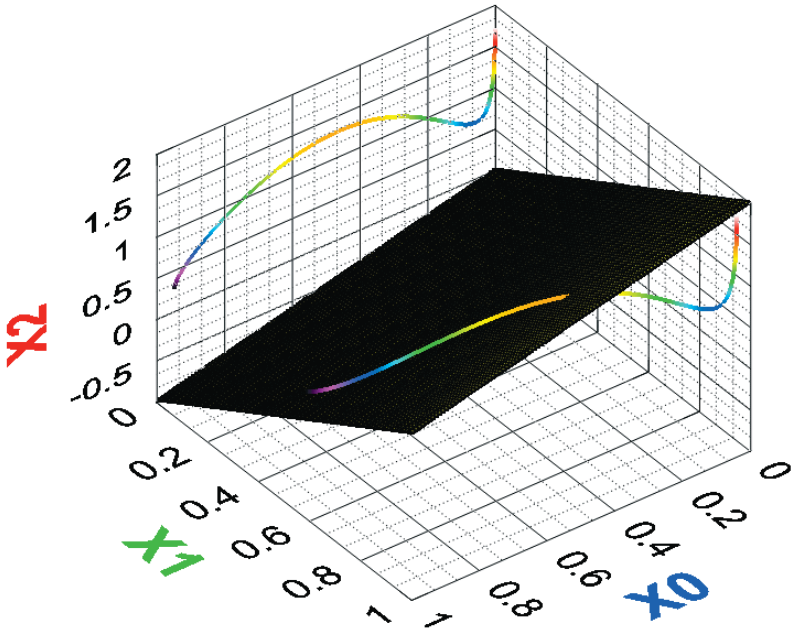


Figure 6.3: The representative curve assuming three samples. The solution to the Peak-Finder equations is equivalent with fitting the curve with a plane. Since the representative curve is not *co-planar* this gives a (small) systematic error. The maximum systematic error correspond to the maximum distance between the curve and the plane.

Notation In the subsequent discussion we shall make an abstraction and deal with an signal of unit amplitude denoted $\boldsymbol{g}$ in order not to make confusion with a measurement $\boldsymbol{m}$. $\boldsymbol{g}$ is used just as a tool to simplify the mathematical notation. Furthermore it is useful to assume $\boldsymbol{g}$ to be a sum of a deterministic sequence $\boldsymbol{y} = [y_0, y_1, \cdots, y_{N-1}]$ and and a stationary random process $\boldsymbol{v} = [v_0, v_1, \cdots, v_{N-1}]$ with $E\{v_i\} = 0; \quad$ for all $\quad v_i \in \boldsymbol{v}$ as depicted in Figure G.1. That is

$$\boldsymbol{g} = \boldsymbol{y} + \boldsymbol{v}. \tag{6.3}$$

Here $\boldsymbol{y}$ and $\boldsymbol{v}$ is statistically independent since $\boldsymbol{y}$ is deterministic.

Furthermore we shall consider the signal evaluated withing a given time range $t \in [t_1, \cdots, t_2]$ denoted the *initial guess range*. We denote this range as $\Delta_t = [t_2 - t_1]/T_s$. The PF vectors are evaluated within this range and only produces valid estimates if the user application is able to guess $\hat{\Delta_{t_0}}$ to a precision of Δ_t. That is, the PF guarantees the systematic error to be smaller than a given maximum value if $t_0^{guess} \in [t_1, \cdots, t_2]$. If the

guess is outside this range then the systematic error might be larger.

The systematic errors as a function of the initial guess range, an other parameters are discussed further in Appendix G. It can be shown that for TOF the algorithm gives a systematic error typically less than $\sim 10^{-4} \cdot \Delta_t$. For amplitude the systematic error is negligible.

The guess range Δ_t is divided in M sub-intervals where M $\geq$ N to give M sub-intervals of length $\varepsilon = \Delta_t/M$. For a given amplitude A, the family of point for different time-delays ε constitutes a curve in N dimensional measurement space (see Section G.1.1 denoted the *representative curve* in the. Figure 6.3 illustrates the special case when N =3. The solving the Peak-Finder equation is in this case equivalent to fitting a two dimensional plane to the representative curve in $\mathbb{R}^3$.

6.2.1.1 Amplitude Estimation with the Peak Finder Algorithm

In the following discussion we shall use the notation $\hat{g}(nT_s + k\varepsilon) = \hat{g}(n)_{k\varepsilon}$ for a time-delay $\varsigma = k\varepsilon$ where $\hat{g}(n)_{k\varepsilon}$ can be any model with a well defined maximum. If the model is the PHOS G2 model we have

$$\hat{g}(n)_{k\varepsilon} = F(nT_s + k\varepsilon)_{A=1} = 1^2\big(\frac{n - k\varepsilon}{20}\big)^2 \exp\big(-2\frac{n - k\varepsilon}{20}\big) + \epsilon(n). \tag{6.4}$$

The PF algorithm implies finding a set of weights $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_{N-1}]$ such that for a signal with unit amplitude

$$E\left\{\begin{bmatrix} \hat{g}(0)_0 & \hat{g}(1)_0 & \cdots & \hat{g}(N-1)_0 \\ \hat{g}(0)_\varepsilon & \hat{g}(1)_\varepsilon & \cdots & \hat{g}(N-1)_\varepsilon \\ \hat{g}(0)_{2\varepsilon} & \hat{g}(1)_{2\varepsilon} & \cdots & \hat{g}(N-1)_{2\varepsilon} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{g}(0)_{(M-1)\varepsilon} & \hat{g}(1)_{(M-1)\varepsilon} & \cdots & \hat{g}(N-1)_{(M-1)\varepsilon} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N-1} \end{bmatrix}\right\} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{6.5}$$

where E denotes the expectation value and $g(n)_\varepsilon = y(n)_\varepsilon + v(n)$. Since $y(n)$ is deterministic and $v(n)$ has zero mean, taking the expectation value yields

$$E\{g(n)\} = E\{y(n) + v(n)\} = y(n) \tag{6.6}$$

so that Equation 6.5 can be written

$$\begin{bmatrix} \hat{y}(0)_0 & \hat{y}(1)_0 & \cdots & \hat{y}(N-1)_0 \\ \hat{y}(0)_\varepsilon & \hat{y}(1)_\varepsilon & \cdots & \hat{y}(N-1)_\varepsilon \\ \hat{y}(0)_{2\varepsilon} & \hat{y}(1)_{2\varepsilon} & \cdots & \hat{y}(N-1)_{2\varepsilon} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{y}(0)_{(M-1)\varepsilon} & \hat{y}(1)_{(M-1)\varepsilon} & \cdots & \hat{y}(N-1)_{(M-1)\varepsilon} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{6.7}$$

This can be written more compact in matrix form as

$$\boldsymbol{g} \cdot \boldsymbol{\alpha} = 1 \tag{6.8}$$

Multiplying both sides of Equation 6.8 with the amplitude A, gives back Equation 6.1 and a correct estimate for the amplitude.

The set of equations given by 6.7 is an overdetermined system (for M¿N) that cannot be solved exactly. The PF is therefore biased, however this bias is very small provided that good initial guess of the amplitude can be provided (Please refer to Appendix G).

6.2.1.2 Timing Estimate by the PF

The PF algorithm for timing implies finding a set of coefficients β such that for a signal with unit amplitude:

$$E\left\{\begin{bmatrix} \hat{g}(0)_0 & \hat{g}(1)_0 & \cdots & \hat{g}(N-1)_0 \\ \hat{g}(0)_\varepsilon & \hat{g}(1)_\varepsilon & \cdots & \hat{g}(N-1)_\varepsilon \\ \hat{g}(0)_{2\varepsilon} & \hat{g}(1)_{2\varepsilon} & \cdots & \hat{g}(N-1)_{2\varepsilon} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{g}(0)_{(M-1)\varepsilon} & \hat{g}(1)_{(M-1)\varepsilon} & \cdots & \hat{g}(N-1)_{(M-1)\varepsilon} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{N-1} \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon \\ 2\varepsilon \\ \vdots \\ (M-1)\varepsilon \end{bmatrix}\right\} \tag{6.9}$$

By taking the expected value we obtain similarly as for the energy

$$\begin{bmatrix} \hat{y}(0)_0 & \hat{y}(1)_0 & \cdots & \hat{y}(N-1)_0 \\ \hat{y}(0)_\varepsilon & \hat{y}(1)_\varepsilon & \cdots & \hat{y}(N-1)_\varepsilon \\ \hat{y}(0)_{2\varepsilon} & \hat{y}(1)_{2\varepsilon} & \cdots & \hat{y}(N-1)_{2\varepsilon} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{y}(0)_{(M-1)\varepsilon} & \hat{y}(1)_{(M-1)\varepsilon} & \cdots & \hat{y}(N-1)_{(M-1)\varepsilon} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{N-1} \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon \\ 2\varepsilon \\ \vdots \\ (M-1)\varepsilon \end{bmatrix} \tag{6.10}$$

Multiplying both sides of Equation 6.9 with constant amplitude does not yield the correct result for the time delay. The time delay is therefore not an invariant of the representative curve. However one can make the time delay quasi invariant by normalizing to unit amplitude. The estimate of the time-delay Δ_{t_0} (in units of (Δ_t)) is therefore found by (since g is defined with amplitude 1)

$$\hat{\Delta_{t_0}} = \frac{1}{\hat{A_{adc}}} \boldsymbol{g} \cdot \boldsymbol{\beta} = \frac{\boldsymbol{m} \cdot \boldsymbol{\beta}}{\boldsymbol{m} \cdot \boldsymbol{\alpha}} \tag{6.11}$$

which in real time amounts to

$$t_0 = T_s \Delta_{t_0}. \tag{6.12}$$

6.2.1.3 Implementation

The extraction of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ vectors was performed by solving Equation 6.7 and 6.10 numerically, with a simulated gamma-2 signal (assuming a deterministic signal) as input. This was done by matrix inversion using the standard linear algebra package in LABVIEW [2] The output of the program is a set of two Peak-Finder vectors. One vector for energy and one for peak position.

An alternative method for extracting the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, with statistically optimized weight coefficients, using the autocorrelation of the measured signal, is described in Appendix G.1.5.1, but is not implemented at present. Coefficient vectors computed this way are expected to give results very similar to those obtained with the present method.

When used in the HLT these vectors are stored in the PHOS specific part of the HLT configuration database and loaded into the *AliHLTPHOSRawAnalyzerPeakFinder(Component)* at start of run.

The most laborious task in the implementation of the Peak-Finder method is to create the program that generates the vectors.

The evaluation of amplitude and peak-position, once these vectors are known, is trivial, as seen from Equation 6.2 and 6.1. The code actually performing this task in HLT is shown in Figure 8.5.

6.2.2 Summary and conclusion

A different approach for the evaluation of amplitude and peak position , the Peak-Finder algorithm, has been presented. Benchmarking results are presented in Appendix G. The variance of the parameter estimates is the same as for the LMS algorithm. The systematic errors are negligible. Since the LMS is an optimal estimator, this means that the the Peak-Finder is also an optimal estimator.

The Peak-Finder algorithm gives optimum parameters estimates 2-4 orders of magnitude faster than fitting techniques (see Appendix G).

6.3 Clusterization

The object of the clusterization is to associated *clusters* of fired cells to single particles.

A detailed description of the clusterizing algorithm used during the offline reconstruction is given in [14] and is only described briefly here.. In addition a faster clusterizer version was developed in order to satisfy the performance requirements of the HLT. The main difference is how overlapping showers are deconvoluted (Section 6.3.1).

[2]LABVIEW is a commercial software package from National Instruments where the programming is done graphically by clicking and dragging symbols. The package includes and extensive mathematical library however most standard mathematical packages could have been used for this purpose.

In the first step a selection cut on the energy, typically 10 - 20 MeV, is applied to all the hits. The hits with energies above threshold are sorted according to their energies in accenting order. The algorithm then starts with the hit with the highest energy and looks for neighbouring channels fired and if found the channels marked as used. A cluster is formed when there are no more neighbouring channels to the most energetic hit. The algorithm then takes the most energetic hit not already marked as used and makes a new cluster. The algorithm is finished when all hits above threshold is associated with a cluster.

6.3.1 Deconvolution of Overlapping Showers

It can happen the two or more particle hits PHOS close enough to each others that the resulting showers are overlapping, thereby creating clusters with several local maxima. This is true in particular for Pb+Pb collisions. The following scenarios should be considered [14].

- Identification of single electromagnetic showers and miss-identification of single hadronic ones as photons
- Recognition and unfolding of two or more overlapping electromagnetic showers
- Recognition of overlap between an electromagnetic and a hadronic shower

During reconstruction showers are a-priori assumed to be electromagnetic. They are deconvoluted by fitting the local maxima with the known lateral distribution of electromagnetic showers. If the shower is hadronic the shower shape will fluctuate wildly from shower to shower and cannot be parametrized. The shower can then be identified as a hadronic one for instance by checking for goodness of fit to the electromagnetic model.

For cells that overlap, i.e is fired by more than one particle a fractions of the energy of the given cell is split between the particles based on the result from the fit.

For the HLT it was found that the deconvolution of showers by fitting will be to slow. It is at the time writing not completely clear how the deconvolution will be performed in the HLT. The use of neural networks has been investigated but no firm results obtained so far. Another option considered is to avoid deconvolution altogether and form pseudo-clusters. The deconvolution can then be done off-line.

6.4 Extraction of Energy and Time of Flight

Once the showers are unfolded and assigned to individual particles the total energy of the particle is found by summing the cells of the shower.

The TOF was so far in simulations taken to be the TOF of the channel containing the most energy, but it should be foreseen to estimate the TOF with a weighted sum of all the cells fired.

6.5 Photon identification

The identification of photons is performed by identifying a particle as something else than a photon. The particles left are assumed to be photons. A detailed description of particle identification in PHOS can be found in [14] and [28] and will only be repeated briefly here.

6.5.1 Discrimination against charged particles

The discrimination against charged particles will be performed by the CPV by correlating hits in the CPV with clusters in the EMC. For charged hadrons shower topology analysis will be used in addition.

6.5.2 Identification of neutral hadrons

In a HI environment hadrons will enter PHOS mainly at MIP energies. Hadrons creating a MIP signal can be rejected by applying a energy cut of ∼200MeV.

The probability that a hadron entering PHOS with MIP energy creates a hadronic shower depends on the exact energy and the particle type. Neutral hadrons creating a hadronic shower can be discriminated against photons, e^+ and e^- by shower topology analysis. Figure 6.4 and 6.5 shows a typical electromagnetic and hadronic shower respectively. As seen in the figures a hadron shower is much wider than an electromagnetic one.

6.5.3 Discrimination against $\bar{n}n$ annihilation

An anti-neutrons entering PHOS at low energy can annihilate a neutron in the $PbW0_4$ crystal thereby creating electromagnetic shower of ∼2 GeV corresponding to the rest mass of the two nucleons. The discrimination between a photon at $\sim 2GeV$ and $\bar{n}n$ annihilation can only be done by measuring the TOF from the collision point [28].

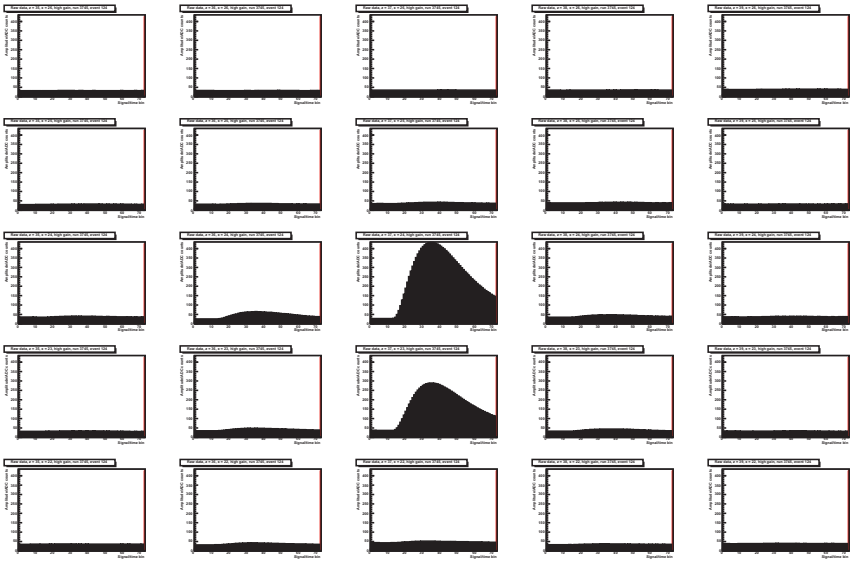


Figure 6.4: Electromagnetic shower induced by a 5 GeV electron. (CERN PS, 2006)

6.6 Measurement of Geometrical Position

The most straightforward method to estimate the coordinate of an incident particle is to calculate the centre of gravity of the shower

$$(z, x)_{rec} = \frac{\sum w_{i,j}(z_i, x_j)}{\sum w_{i,j}} \tag{6.13}$$

where (z_i, x_j) is the cell coordinate of channel i relative to the channel containing the largest fraction of the energy and $w_{i,j} = E_{i,j}/E_{total}$.

The position resolution can be parametrized as

$$\sigma_{z,x} = \sqrt{A_{z,x}^2 + \frac{B_{z,x}^2}{E}} \tag{6.14}$$

The improvement of $\sigma_{z,x}$ for larger energies comes from the fact that more cells are fired at higher energies.

The values of A and B depends on the angle of the imping particle relative to the PHOS detection plane.

Some values, based on simulations [28], are given in Table 6.1.

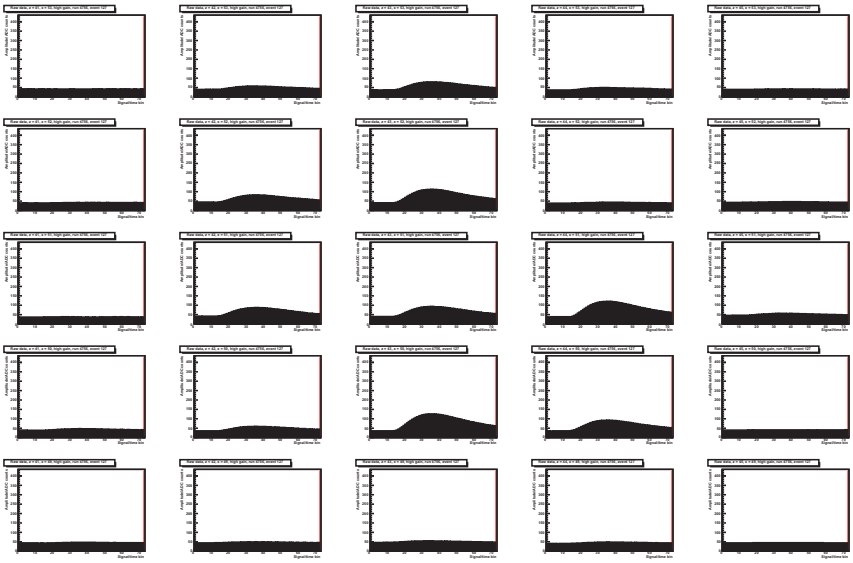


Figure 6.5: Hadronic shower resulting from irradiation of 5 GeV electrons on a copper target. (CERN PS 2006)

α (degrees)	$A_{z,x}/cm$	$B_{z,x}/cm \cdot GeV^{\frac{1}{2}}$
0°	0.032	0.264
3°	0.070	0.231
6°	0.147	0.170
9°	0.198	0.155
Average, all angles	0.096	0.229

Table 6.1: $A_{z,x}$ and $B_{z,x}$ as a function of the incidence angle.

Chapter 7

Calibration

The conversion from ADC amplitudes to energy must be both relatively and absolutely calibrated. Relative calibration means that any two detection channels seeing the same particle with the same energy hitting at the same place on the crystal should see the same energy after conversion from the ADC amplitude. Absolute calibration means that the absolute energy scale is correct. For instance the electromagnetic shower induced by a 2 GeV electron should amount to exactly 2 GeV when all fired cells are summed up.

One of the contributions to systematic errors is the determination of the absolute energy scale. Although the energy scale can be reasonably well determined at energies up to a few GeV the measurements at higher energies suffer from non-linearity effects and low statistics. Non-linearity effects were measured at CERN PS in 2006 using 1 - 5 GeV electron beam [51]. Non-linearity effects of up to 5% were found for some crystals with large variations between different crystals. For higher energies the non linearity-effects were studied using simulations [51].

Several methods for calibration have been considered. Further information on PHOS calibration strategies can be found in [51]. It is concluded that no single method will be sufficient but that PHOS should be calibrated using a combination of several methods. The various methods considered are summarized below.

- Equalization of energy distribution: to the first approximation all the detection channels see the same physics so the energy distribution should be the same in all channels.

- Electron beam: direct absolute and relative calibration by using electrons of known energy

- Cosmic MIPs.

- MIPs during collisions.

- π^0 and η invariant mass globally on detector level: relative calibration by minimization of π^0 width, absolute calibration by π^0 and η peak position
- π^0 and η invariant mass for individual channels: absolute and relative calibration by π^0 peak position
- Track matching with TPC: matching of electrons detected with TPC with hits in PHOS, using the momentum obtained from the TPC to calibrate PHOS.

Simulation studies [51] [52] have been performed to investigate several of the options listed above. For real data, calibration with electron beams and with cosmic muons were used so far. These two methods are therefore the main focus of this chapter. In addition it was demonstrated at CERN PS in 2003 that PHOS is capable of measuring π^0 and η masses with a good mass resolution. The results are shown in Figure 4.13.

7.1 Calibration Issues

The Gain Dependence on V The gain as function of the bias voltage can be described by the formula

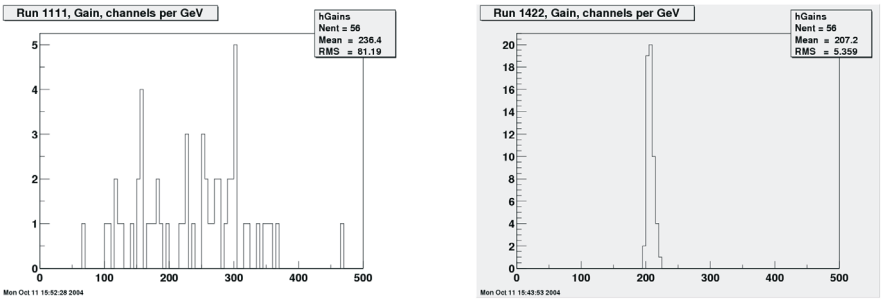
$$M(V) = p_0 + p_1 \exp(p_2 V) \tag{7.1}$$

where $p_0 \approx 1$ and p_1 and p_2 is given by the intrinsic gain factor of the individual APD which can vary considerably from APD to APD. Typical values are $p_1 \approx 0.01$ and $p_2 \approx 0.02$ [29]. As seen from Equation 7.1 p_0 dominates for a reverse bias less than $\sim$ 100 V, giving a gain of M=$\sim$ 1 (i.e. no avalanche multiplication). For higher voltages p_1 and p_2 dominates, giving an exponential increase of the gain for increasing voltage.

Spread of the APD Gains Figure 7.1(a) shows the response in ADC counts for a 2 GeV electron for a collection of 56 channels when the DAC setting is set to 512, corresponding to $\sim$300V for all channels. As seen in the figure the spread in intrinsic gains is more than 500 %. Figure 7.1(b) shows the gains after applying calibrated DAC setting obtained by electron beam calibration. As seen in Figure 7.1(b) the spread in gains after applying calibrated DAC settings is $\sim$2.5 %.

The Lower Limit of the Precision The best possible calibration that can be obtained by tuning the HV is given by the slope of the gain curve at a target gain and the granularity of the DACs. The gain curve was measured precisely for some APDs [13] used for EMCAL (EMCAL use gain 30 for the APDs) and found to be

$$\frac{1}{M}\frac{dM}{dV} = \frac{2.3\%}{V} \qquad \text{at} M = 30 \tag{7.2}$$



(a) No calibration, HV = ∼300 V

(b) Calibrated DAC settings

Figure 7.1: Left:the spread of APD gains without calibration. Right: The spread of APD gains after applying calibrated DAC settings.(Data taken in October 2004).

at 22 °C. The 10 bit DACs regulates the HV in the range 210 - 400 V with a granularity of $(400-210)/1023 = 0.1857\ V/bit$ thereby limiting the best possible relative calibration attainable by tuning of the DAC settings to 0.46 % at M = 30. It should be noted here that since PHOS will be operated at M = 50 the slope will be bigger as seen from Equation 7.1. Consequently the best attainable equalization of gains will be worse.

Temperature Dependence of the Light Yield The temperature coefficient of the light yield of the $PbWO_4$ crystal is $-2\%/C^0$ for a broad temperature range. At a temperature of -25°C the light yield of the crystals is three times higher than at room temperature. In addition the gain of the APDs increases and the noise decreases at lower energies. All in all these factors lead to a better energy resolution at low energies when the crystals are cooled down. It was therefor decided [14] that the detection channels of PHOS should be operated at -25 ± 0.3°C.

Temperature Dependence of Gain The APDs are semiconductor devices with a gain factor that is strongly dependent on temperature. The stabilization of the temperature is therefore also necessary in order to have a constant gain of the APDs. The temperature dependence is shown in Figure 7.4. As seen in the figure the gain decreases with with increasing temperature and the dependence is approximately linear. The temperature dependence is stronger for a higher gain. At M = 50. the temperature dependence is:

$$\frac{1}{M}\frac{dM}{dT}|_{M=50} \approx \frac{2.2\%}{^\circ C}. \tag{7.3}$$

The APDs arrive from the manufacturer with data-sheets for each APD specifying in particular the reverse bias voltage required to obtain an electron gain factor of 50 at room

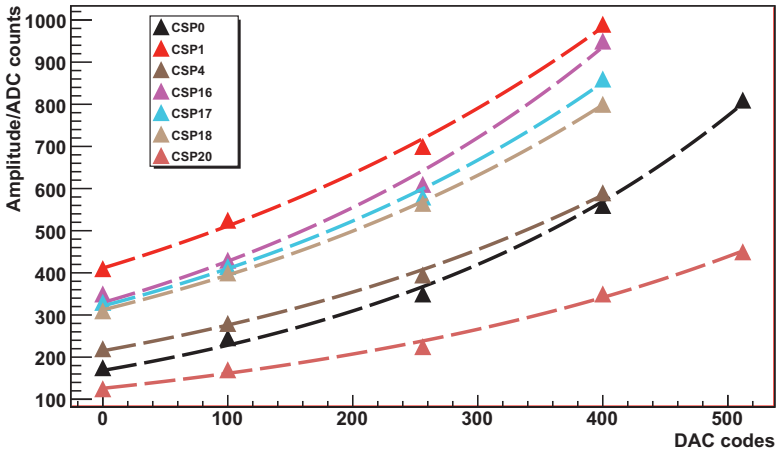


Figure 7.2: Amplitude as a function of the DAC settings for 7 different detection channels. The measurements are indicated by the triangles. The stitched line is the fit result using the formula $A = p_0 + p_1 \exp p_2 DAC$.

temperature. This is the nominal gain factor for PHOS however since PHOS is designed to be operated at -25°C it is necessary to extrapolate the HV down to this temperature from the value given in the data-sheet.

Non Linearity Effects The scintillation light is attenuated by its passage through the $PbWO_4$ material. A detailed study of the optical properties of the $PbWO_4$ crystals is presented in [14]. A simplified model used for instance during simulation in AliRoot estimates the mean number, $< N_\gamma >$, of scintillation photons collected by the APD with the formula

$$< N_\gamma >= E_d \tilde{N}_\gamma \epsilon_{APD} \exp(-Ad), \tag{7.4}$$

where $\tilde{N}_\gamma = 4.7x10^4 \gamma GeV^{-1}$ is the average number of scintillation photons per deposited energy, $\epsilon_{APD} = 0.0266$ is the photo-efficiency of the APD, $A \approx 0.0045$ is the attenuation factor[1], d is the distance between the *hit* to the APD and E_d is the energy deposited at distance d from the APD.

The definition of a hit given in [53] is *"The transport package transports the particles through the set of detectors,and produces* ***hits****, which in ALICE terminology means energy deposition at a given point"*. For PHOS the photon creates an electromagnetic shower and

[1]The attenuation can vary from crystal to crystal

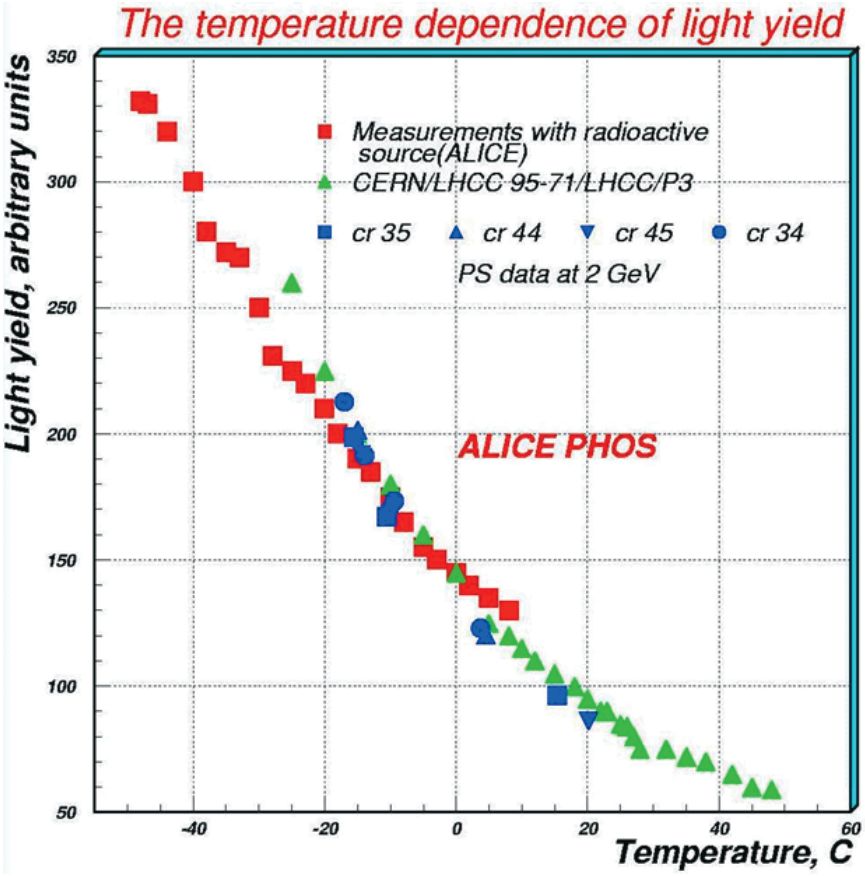


Figure 7.3: Light yield of the $PbWO_4$ crystals as a function of the temperature.

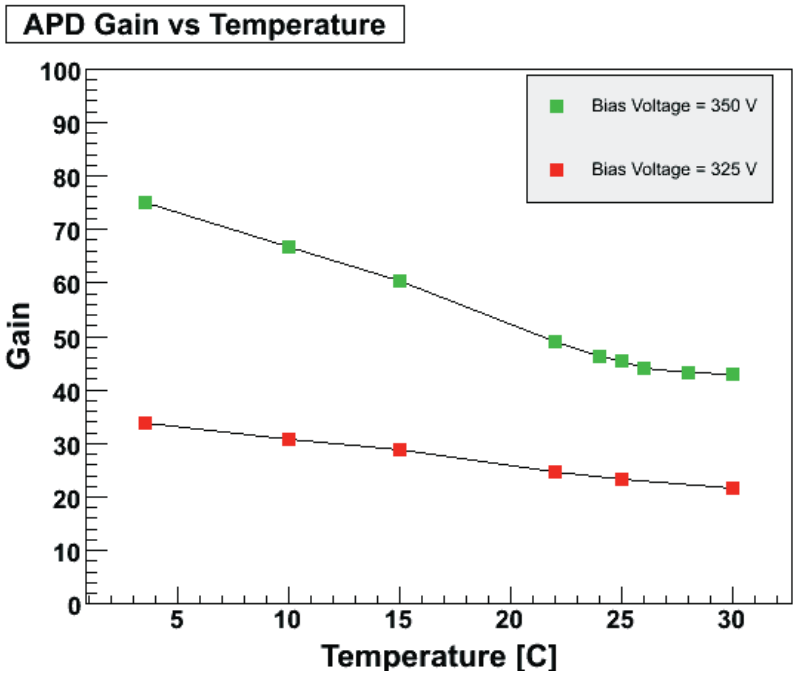


Figure 7.4: Gain of the APD as a function of the temperature [13].

thereby many hits. We shall therefore instead introduce d_{eff} which is the mean distance between hits produced by single particle to the APD.

The lower the energy of the incident particle the closer to the detection surface of the crystal the electromagnetic shower ceases to develop. The scintillation light is therefore attenuated more.

For very high energies > 10 GeV some of the particles createt in the shower exites the back end of the crystal with an energy higher than E_c. This is called *shower leakage*.

In addition systematic errors can be introduced during the reconstruction when a threshold is applied on the signal from individual readout channels prior to the clusterization. The combined effect of the attenuation, the shower leakage and the clusterization was simulated [51]. Figure 7.5 shows some of the results. The most realistic scenario is the results obtained with attenuation and a single channel energy cut of $E > 20MeV$ prior to clustrization indicated by the black diamonds. As seen in the figure the ratio $E_{measured}/E_{real}$ is lowest for the smallest simulated energy of 100 MeV because of the combined effect of both attenuation and the systematic error introduced by the clustering threshold being at its maximum. Above $\sim$ 30 GeV the the ratio drops off because of shower leakage.

7.2 Methods

The nominal detector response is given by the requirement that the individual high gain readout channels should have a dynamic range of 5MeV - 5GeV whereas the high gain readout channels should have a dynamic range of 80 MeV - 80 GeV. Since a particle hitting the center of one crystal deposits approximately 80% in this channel this gives a total dynamic range of 6 MeV - 100 GeV in detecting particles where the high gain channel can detect particles up to 6.25 GeV. The goal of the calibration is in general to ensure that these requirements are satisfied as accurately as possible for all detection channels. The calibration parameters are evaluated for the high gain channels. Calibration parameters for the low gain channels are obtained by measuring the HG/LG gain ration for instance by using the LED monitoring system.

The calibration can be categorized in two dimensions, method wise in *relative* and *absolute* calibration, and detector-vise in *Hardware* and *software* calibration as given in Table 7.1.

Calibration at the *hardware* level means calibration obtained by the configuration of the PHOS front-end electronics. This concerns mainly the reverse bias voltage of the APDs, which is controlled indirectly via the FEE board controller and the digital to analogue converters situated on the front-end cards. In addition the configuration of the ALTRO chips by means of zero suppression thresholds and equalization of baselines could

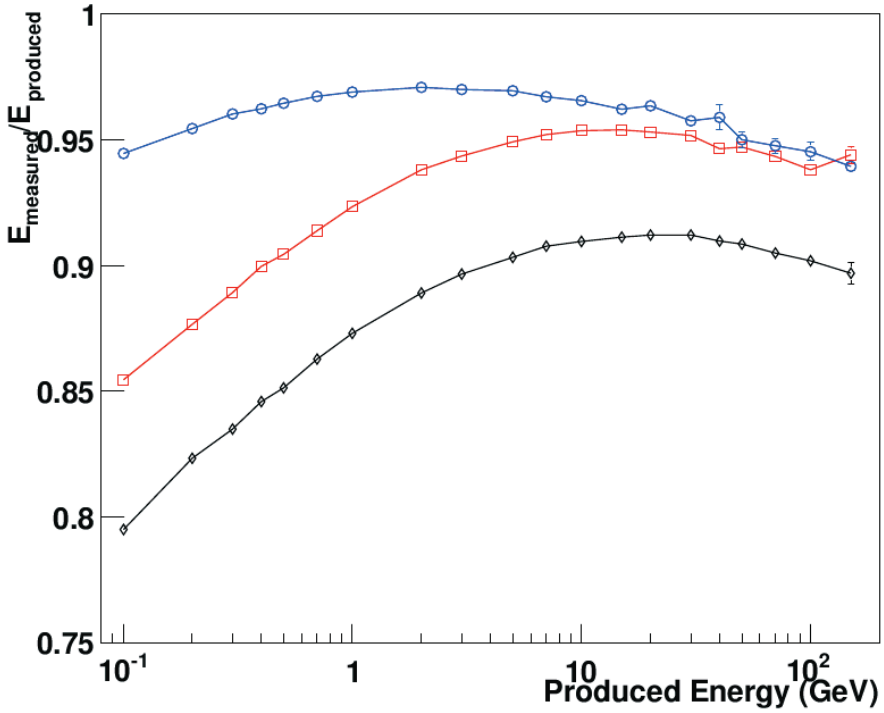


Figure 7.5: Ratio of measured cluster energy to the incident photon energy for various parameters used during the simulation and the reconstruction. Blue circle: No attenuation and no threshold. Red squares: No attenuation with threshold (4 ADC counts for single channels). Black diamond: With Attenuation and threshold. [51].

	Hardware	Software
Relative	APD bias control	Electron beam analysis Minimization of π^0 and η width Equalization of energy distribution MIP
Absolute	APD bias control	π^0 and η peak position Electron beam analysis MIP

Table 7.1: Method vs detector view of calibration.

in principle also be considered a part for the calibration, but is not considered in this chapter.

Calibration at the *software* level means that the conversion factor used during on-line or off-line analysis in order to convert from signal amplitudes in entities of ADC counts to real energy is tuned in order to reconstruct the energy of real particles as accurately as possible.

A good *relative* or *inter channel* calibration means that if a particle of a given energy hits different detection channels (crystals) then the energy output of these different channels are the same. It was demonstrated during the electron bean test at CERN PS in 2006 that a relative calibration at approximately 3% (Section 9.1.1) can be achieved by the configuration of the APD bias voltages alone which is probably more than sufficient for the L0 trigger efficiency, however not sufficiently accurate for the analysis. During analysis an imperfect relative calibration affects the constant term of Equation 4.6. Ideally it should be small enough compared to the other two terms Equation 4.6 to be ignored for all energies.

A proper *absolute* calibration means that the energy scale is correct. In particular an electron with a known energy of 2 GeV should be observed as 2GeV particle in the detector and not as a particle with some other energy. In addition to electron beam calibration it has been proposed to use the invariant mass method on π^0 and η to establish the correct absolute calibration/energy scale for PHOS. The advantage is tree fold redundancy in obtaining the correct energy scale.

7.3 Calibration with Electron Beam

The most accurate calibration is performed by using an electron beam with known energy. The electrons, since they create electromagnetic cascades, deposit a signal identical to that of a photon. The energy of the beam is generally known with a much better precision than the anticipated resolution of the detector. This has the additional advantage that electron beam tests can also be used for direct measurements of the energy resolution. Figure 7.6 show schematically the general setup used during beam tests at CERN PS. A similar setup was used also during beam tests at CERN SPS but included in addition a copper target for the measurements of light mesons. The trigger setup consists of six plastic scintillator triggers S1, S2, Sz, Sx, S3 and S4 and a Cerenkov gas detector. S1 is situated close to the output of the beam. S2 and S3 are positioned on each side of the Cerenkov gas detector. The Sz and Sx *fingers* comprise two thin scintillators orientated perpendicular to each other in the detection plane and gives the possibility to control

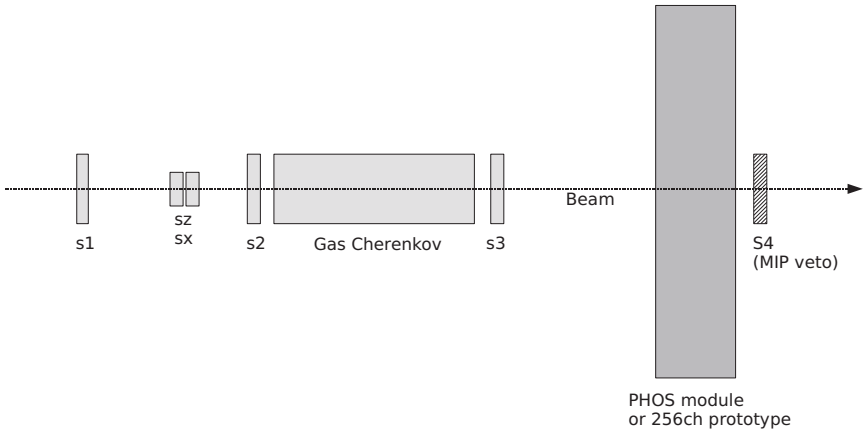


Figure 7.6: Setup during electron beam tests at CERN PS with the PHOS 256 channel prototype. The beam direction is from left to right. Trigger is issued on coincidence between the plastic scintillator triggers S1, S2, Sz, Sx, S3 and trigger from the gas Cerenkov counter situated between S1 and S2. The Sz and sX *fingers* pinpoints the beam to trigger on a limited area of the detection plane of the PHOS module/prototype. For some of the beam tests there was an additional MIP veto scintillator (S4) situated behind the PHOS module/prototype.

the wideness of the beam trigger (the overlap area of the fingers is typically a few square centimeter).

A trigger is issued on coincidence between S1, S2, Sz, Sx, s3 and S4 and the Cerenkov gas detector. For some of the beam tests of the prototype there was in addition a MIP veto trigger situated behind the detector in order to increase the electron trigger efficiency.

The electron beam is always steered at the same position. Different areas of the detector is irradiated with the beam by moving the PHOS module on an *z-x frame*. Moving in the z (horizontal) direction is done physically inside the beam area and requires veto of the electron beam and people with the appropriate permission at shift to go inside the beam area to move the module. Movement in the x (vertical) direction is controlled remotely from the control room. The trigger is typically configured to select electrons hitting PHOS within an area of roughly 10x10 cm^2. The relatively wide trigger area (controlled with the Sz AND Sx fingers) is used in order to increase the trigger rate. Since a single crystal covers an area of 2.2x2.2 cm^2 this means that the position of a hit relative to a crystal is not *a-priory* known on an event by event basis, but the position of the hit can be estimated by evaluating the shower profile.

Figure 7.7 show a typical single channel raw amplitude spectrum resulting from irradiation of the first PHOS module with 2 GeV electron beam at CERN PS in 2006. Before continuing the discussion we should introduce the term *edge* or *electron edge* which will

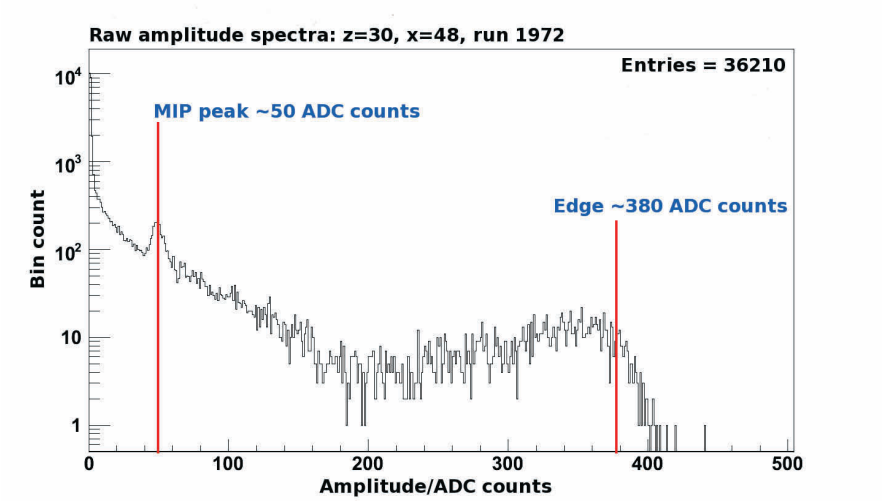


Figure 7.7: Typical raw amplitude spectrum for a single channel resulting from irradiation with 2 GeV electron beam. The data was taken during the electron beam calibration with the first PHOS module at CERN PS in 2006. The most important information to be extracted from the spectrum is the position of the edge. It will in addition it has also been considered to use the MIP peak for calibration.

be used interchangeably in this chapter. The term is somewhat misleading since it the electron edge is not really an edge. Nevertheless since this term was used internally in PHOS for some time when describing the extraction of the calibration coefficients from an amplitude spectra, such as for instance the spectra shown in Figure 7.7, it will be used also here. We introduce following definition:

Definition 1 *The* electron edge *is, given a raw amplitude spectrum, the histogram bin filled by an electron hitting exactly in the center of a crystal, i.e it is the amplitude in ADC counts seen when the electron deposits 80% of its energy in a single crystal.*

The most important information to be extracted from the spectra is the position of the electron edge as indicated in Figure 7.7. The shape of the amplitude spectra is a convolution of several effects, mainly the wideness of the beam and the shower profile of a central hit by an electron. Neighbouring crystals absorbing fringe parts of the electromagnetic cascades, contribute to the low-amplitude part of the spectrum.

The shape of the edge depends on parameters such as the exact position of the irradiated crystal with respect to the beam and the quality of the beam etc and cannot be described by a general analytical formula. The determination of the edge from the raw spectra is therefore the main source of systematic errors in the calibration. The edge finding algorithm used for PHOS during beam tests in 2003 - 2006 uses an ad hock approach

based on experience from previous beam tests [54].

It is important to point out here that the edge is in general not the the peak position of the shoulder that would be found for instance by fitting the upper range of the amplitude spectra with a Gaussian.

The flux of minimum ionizing muons is also quite large and as shown in the figure there is a clear MIP peak at ~ 50 ADC counts. The MIP signal can be used as complementary information but the signal is too low to be used as the only signal for calibration.

According to the requirements discussed in Section 7.2, a central hit by a 2 GeV electron, depositing 80 % of its energy in that channel, should give an amplitude of 320 ($400 \cdot 80\%$) ADC counts in that channel.

Calibration can of course also be done using other beam energies, however 2 GeV was the energy that was most often used so far and at this energy the aim of the calibration is to position the edge at 320 ADC counts by tuning the APD reverse bias voltage.

Figure 7.7 shows the amplitude spectrum only for a single run and a single channel however typically a series of several runs is taken. One starts out with a given APD high voltage configuration and irradiate as many channels as possible accumulating as much statistics as possible, determined by the available beam time, without changing the HV settings. The trigger is configured for a relatively narrow beam, so that most of the triggered electrons typically hit within a 2x2 or 3x3 matrix of crystals. In order to reduce the data volume to a minimum the electronics is configured to only read out detection channels surrounding the position of the beam, most often either a 5x5 or 7x7 matrix of detection channels.

When a series of runs with a given APD bias configuration is finished the raw spectra from different runs are merged on a per channel basis in order to obtain as much statistics as possible for each individual channel. Edges are found for all of the irradiated channels and the result is used to make a new APD bias configuration. A new series of runs is taken with the new configuration and new edges evaluated, an so on. The number of iterations depends on the effective beam-time available but 2-3 iterations is sufficient to obtain fine tuned APD bias settings.

The energy resolution obtained by tuning the APD HV settings alone is limited by how accurate the electron edge can be extracted from the raw ADC amplitude spectra and is therefore typically worse than the granularity of the DACs (see Section 7.1.

The energy resolution obtained by tuning the APD HV settings is typically in the range $\frac{4-10\%}{\sqrt{E}}$. This is sufficient for the PHOS L0/L1 trigger decision generated by the TRUs but not for off-line and on-line event reconstruction and analysis.

Fine tuned, per channel, conversion factors or *calibration constants* can be obtained in software by finding, for a set of L irradiated crystals, a set of ADC to energy conversion factors $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_L]$ that minimize the difference between known beam energy and measured cluster energy. Mathematically, the minimized quantity is the functional $F(\boldsymbol{\alpha})$

defined as

$$F(\boldsymbol{\alpha}) = \sum_{k=1}^{N}(\sum_{i=1}^{L} \alpha_i A_i^k - E_{beam})^2 \tag{7.5}$$

where k spans over N events, α_i is the conversion from ADC amplitude to real energy for crystal i, A_i^k is the ADC amplitude deposited in crystal i at event k and E_{beam} is the known beam energy [14].

7.4 Calibration with Minimum Ionizing Cosmic Muons

Cosmic rays are energetic particles originating from space and impinging on the earth's atmosphere. Approximately 90 % are protons, about about 9% are alpha particles, and about 1% are electrons.

Cosmic rays with an energy above 10 GeV approach the earths surface isotropically. This is probably due to galactic magnetic fields causing cosmic rays to travel in spiral paths, rather than the sources being uniformly distributed. The cosmic rays interact with nuclei of atmospheric gases, mainly oxygen and nitrogen creating a cascade or "shower" of secondary particles. These showers can span several kilometers in diameter and contain millions of particles.

Most of the secondary particles are unstable mesons, mostly pions and some kaons, which quickly quickly decay into muons. Due to the relativistic time dilation and the fact that muons do not interact strongly the muons are able to reach the surface of the earth. At the earth‘s surface these cosmic showers consists of approximately 75% muons. The muon flux at sea level has a mean energy of 2GeV i.e in the minimum ionizing regime which makes them good candidates for calibration.

The energy deposited in matter by heavy relativistic particles is described by the Bethe-Block function [46] The distribution of deposited energy in a detector material, such as the $PbWO_4$ crystal, is obtained by a convolution of the Bethe Block formula with the energy distribution of vertical cosmic muons (considering only vertical muons). However, to the first approximation all cosmic muons can be regarded as MIPs.

However, as seen from Figure 8.9, the energy distribution has a broad peak around 100 GeV and falls off at higher and lower energies. In this region, the Bethe-Block formula gives a slowly varying energy deposition, close to the minimum ionizing particle value. In the following discussion, it will therefore be assumed that all the muons are in the minimum ionizing regime with deposited energy of

The deposited energy by a cosmic muon in the $PbWO_4$ is [55]

$$\frac{dE}{dx}_{min} = 1.23 MeV g^{-1} cm^2 \tag{7.6}$$

The density of the $PbWO_4$ is $8.28 g/cm^3$, giving a deposited energy in a crystal of $10.17 MeV/cm$. For a muon traversing the crystal vertically the mean deposited energy is

$$E = 10.17 MeV/cm \cdot 18 cm \approx 200 MeV \tag{7.7}$$

The main disadvantage of using cosmic muons for calibration for PHOS in particular is that the energy deposited is very low compared to the dynamic range. A MIP will deposit $\sim$ 200 MeV whereas the dynamic range is 5 MeV - 100 GeV. If non-linearity effects are not fully understood the extrapolation of calibration constants to high energies can lead to large systematic errors in the energy measurements of high energy particles. The advantage is that cosmic ray particles are available anywhere and at any time at earth. Consequently calibration using cosmic rays is much less manpower intensive than for instance calibration using electron beam.

Chapter 8

The High Level Trigger Implementation for PHOS

The physics requirements for detectors participating in the HLT is given in Chapter 3. The implications of these physics requirements for PHOS are given in this chapter.

8.1 Trigger

The main focus for the work on the PHOS HLT so far has been debugging and development of reconstruction / data compression algorithms, for the extraction of amplitude and time-of-flight parameters from PHOS raw data. The physics trigger has not yet been fully implemented or tested.

A prototype of a high p_T trigger was implemented and tested during the commissioning of PHOS in ALICE in September - October 2008. This trigger simply sums the energy deposited in all the detection channels for one PHOS module (the one commissioned in ALICE) and forwards the reconstructed event and the raw data to the PHOS-HLT on-line display. If the sum exceeds a given user defined threshold. This trigger was not yet used as trigger for DAQ.

To trigger on γ-jet signals in HLT as outlined in Section 4.2.3 and 3.3.1 would require the PHOS HLT to detect and trigger on *isolated photons*. Thus the condition for a γ-jet trigger from HLT will be a jet trigger obtained by combining data from EMCAL, ITS, TRD and TPC an isolated photon trigger from PHOS.

8.2 Design and Organization of the PHOS HLT Software

The complete HTML documentation of the PHOS HLT software is available at http://folk.uio.no/perthi/alice/doc/phos_hlt_htmldoc/classes.html. Only the most important components will be described here.

The on-line processing for PHOS can be separated in three levels; *RCU/Equipment level*, *Module level*, and *Detector (PHOS) level*. An additional fourth level, the *ALICE level*, becomes relevant when data from PHOS is combined with other sub-detectors for global event characterization and HLT trigger decision.

The PHOS analysis steps associated with each level is (refer to Chapter 6)

- RCU level

  Extraction of energy and TOF from individual channels

  Accumulation of per channel calibration data

- Module level

  Clusterizing

  Deconvolution of overlapping showers

  PID

  Provide data for the PHOS HLT on-line display

- PHOS level

  Reconstruction of π_0s

  PHOS HLT trigger decision

  Filling of PHOS specific ESD

- ALICE level

  Global event characterization by merging detector-specific ESDs to a single HLT ESD

  Global HLT trigger decision

  Provide data for the global HLT on-line display (by sending the ESD)

The PHOS HLT software exploits the parallel nature of the data taking and the architecture of the HLT to the maximum by performing as much as possible of the processing in parallel. Figure 8.1 illustrates the parallel nature of the data acquisition, the relevant parts of the HLT architecture and the processing steps associated with each level of the DAQ/HLT architecture. As seen in the figure the HLT has a tree-like structure where the

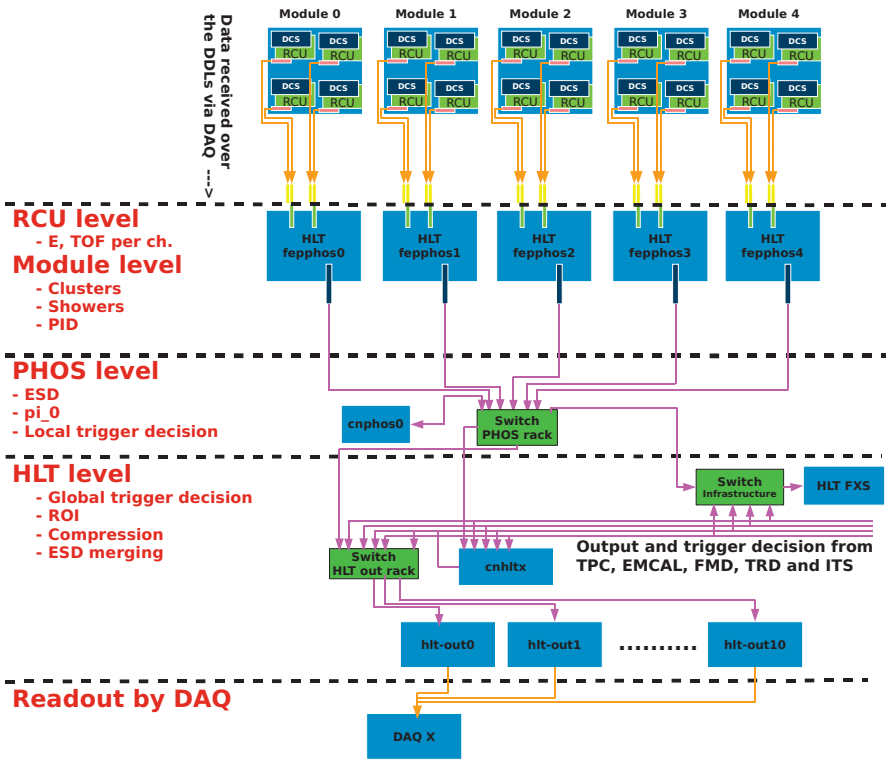


Figure 8.1: Data flow of the on-line processing in the HLT. The figure shows how the distributed nature of the data acquisition and the architecture of the HLT is exploited in order to maximize the performance of the on-line analysis. The data is first processed on the *fep* nodes. The resulting output is merged and processed on the *cnphos* (cn = computing) node. The output from *cnphos0* is forwarded to one of the global HLT computing nodes (denoted *cnhltx* in the figure.)

data is concentrated on fewer and fewer nodes. At the final level, the *HLT level*, a full reconstructed event is available on a single node.

The transportation of the data between each level of the processing is done using one-to-one gigabit Ethernet switches. It is important for the overall performance that the data is compressed as much as possible at each level in order to minimize the network traffic.

As shown in Figure 8.1 the data arrives from PHOS to the HLT, via DAQ (see Chapter 3), distributed over 40 DDLs for the full PHOS detector with five modules. Each module is read out using four DDLS and each DDL reads the data from one equipment.

Each PHOS FEP in the HLT receives data from one PHOS module and consequently receives the data from one module and one event distributed over four DDLS. The RCU and module level processing should therefore, if possible, be performed on the FEPs.

The resulting output data from the processing on the FEPs is forwarded to the PHOS computing node *cnphos0*[1] which has access to the full reconstructed event from PHOS.

The processing at the PHOS level is therefore performed on *cnphos0*. The processing that takes place at each level depends on the configuration of the PHOS HLT processing chain [2]. In the following we shall assume that chain is configured to produce ESDs and to provide a trigger decision.

The *cnphos0* node collects event fragments from the five modules into one PHOS ESD. This ESD is merged with ESDs from other detectors on the next level, the HLT/ALICE level, to form one HLT ESD. The local PHOS trigger decision is also done on the PHOS level, It is foreseen that each participating sub-detector should make its own local trigger decision which is then forwarded to the computing nodes processing the data on the HLT level for a global HLT trigger decision.

The output from the processing at the HLT level is sent to the HLT-out PCs and subsequently read out by DAQ. A single HLT event is always written to a single *HLT-out* node, however, different nodes might be used from event to event in order to achieve load balancing.

8.2.1 HLT Analysis Components

The term HLT component denotes a C++ class, which acts in the HLT as a plug-in and runs and runs as a separate process on one of the nodes.

It takes as input a well defined data format and produces as output a well defined data format but does not care about the producer of the input data or the receiver of output data.

It is foreseen that the functionality, or *business logic*, of the components should be accessible within the AliRoot framework independently of HLT. The analysis components are therefore organized in pairs on the form
AliHLTPHOS<name>.cxx/h and *AliHLTPHOS<name>Componen.cxx/h* where in general the
AliHLTPHOS<name>Componen.cxx/h makes the functionality provided by
AliHLTPHOS<name>.cxx/h available within the HLT.

The name ...< name > class performs the actual analysis and the ...< name >Component class acts as an interface to the HLT framework. There is a strict separation between the

[1] During the commissioning in ALICE the *cnphos0* node was not yet installed but since only one PHOS module was installed one of the unused FEP nodes was used instead.

[2] It is specified in a xml configuration what kind of processing takes place at each level, which processes sends an receive data to other processes, to/from which nodes etc.

business logic and the interface to the HLT. It is forbidden for a class ending with *Component* to analyze any data, and it is forbidden for a class not ending with *Component* to know anything about the interface to the HLT. This separation follows the software design principle denoted *the principle of least privilege* which implies that a C++ object only has access to exactly the data or interfaces that are strictly necessary for its task.

8.2.1.1 HLT Analysis Components for PHOS Raw Data

One of the most important HLT components for PHOS is the *AliHLTPHOSRawAnalyzerComponent* which is used in all PHOS configurations for the on-line processing.

This component, or more precisely, the associated analysis class RawAnalyzer, decodes the data and delegates the evaluation of amplitude and peak position to one of the three derived classes *AliHLTPHOSRawAnalyzerLMSComponent*, *AliHLTPHOSRawAnalyzerPeakFinderComponent* or *AliHLTPHOSRawAnalyzerCrudeComponent*.

The *LMS* and the *Peak-Finder* component evaluates these parameters as described in Section G.1. The *Crude* component uses the maximum ADC value subtracted the baseline as an estimate for the amplitude and the corresponding sample index as the estimate for the peak position.

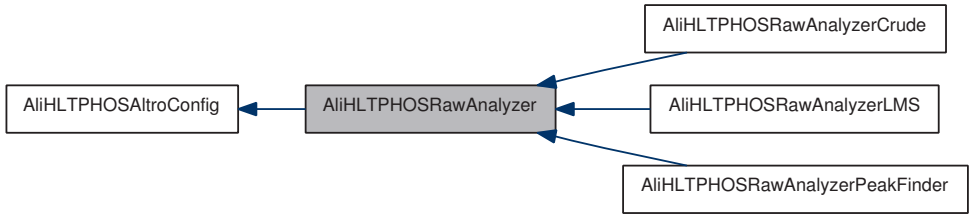
The output, the resulting amplitude and peak position estimates, are pushed to other processes. Typical recipients of this output data is the *..ClusterizerComponent* which does exactly what the name says, or the PHOS HLT on-line display. Furthermore the *RawAnalyzer* can be configured to attached the raw data, something that can be very useful for debugging purposes[3]. Figure 8.8 shows an example when the *RawAnalyzer* is used in this mode.

8.2.1.2 The Implementation of the Peak-Finder Algorithm in the HLT

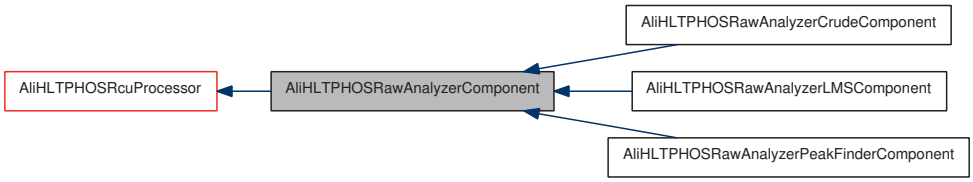
As already mentioned in Section 6.2.1.3 the most difficult part of the implementation is to make the program that generates the vectors. The implementation of the extraction of amplitude and peak position using the Peak-Finder is trivial. Nevertheless, the current[4] implementation of the Peak-Finder is shown in Figure 8.5 (The variable names have been shortened for typographical reasons). Line one and two resets the two variables holding the peak position (fDTof) and energy (fDAmpl). Line four checks that the two Peak-Finder vectors are initialized. Line seven and eight checks if the array of data is longer than the size of the Peak-Finder vectors. If so only a subset of the data corresponding to the length of the Peak-Finder vectors is used. Line nine and 10 evaluates the amplitude and peak position respectively. Line 11 and 12 sets amplitude to the max ADC value (1023)

[3]This feature was extensively used during the commissioning of PHOS.

[4]AliRoot trunk Jan 23 2009



(a) Inheritance diagram for *AliHLTPHOSRawAnalyzer*



(b) Inheritance diagram for *AliHLTPHOSRawAnalyzerComponent*

Figure 8.2: Class inheritance diagram for *AliHLTPHOSRawAnalyzer* and *AliHLTPHOSRawAnalyzer-Component*. The figures were automatically generated by Doxygen using the PHOS HLT source code as input.

if the signal is in saturation. Finally, line 13 normalizes the TOF to unit amplitude (see Section 6.2.1.2).

8.3 Data Rates and Compression

8.3.1 Data Rate

The data rate from PHOS is limited upwards by the bandwidth of the DDL to 200 MByte/s per DDL giving a data rate of 4 GByte/s for the full PHOS detector.

During normal LHC physics running conditions, the occupancy of PHOS is linked to the multiplicity dN/dy per unit of rapidity at mid rapidity. The multiplicity is one of the main unknowns at LHC. Current estimates based on results from RHIC and SPS range between $dN/dy \approx 2000$ and $dN/dy \approx 8000$ most probably in the lower end of this range [15]. In addition the momentum distribution of the radiated particles is also relevant since a high momentum particle fires more cells (detection channels) than low momentum particles.

The occupancy and the average number of samples per hit were measured for simulated Pb+Pb events and for real cosmic data taken at P2. The results are shown in Table 8.1 and 8.2 respectively. The relevant parameter with respect to data compression is the

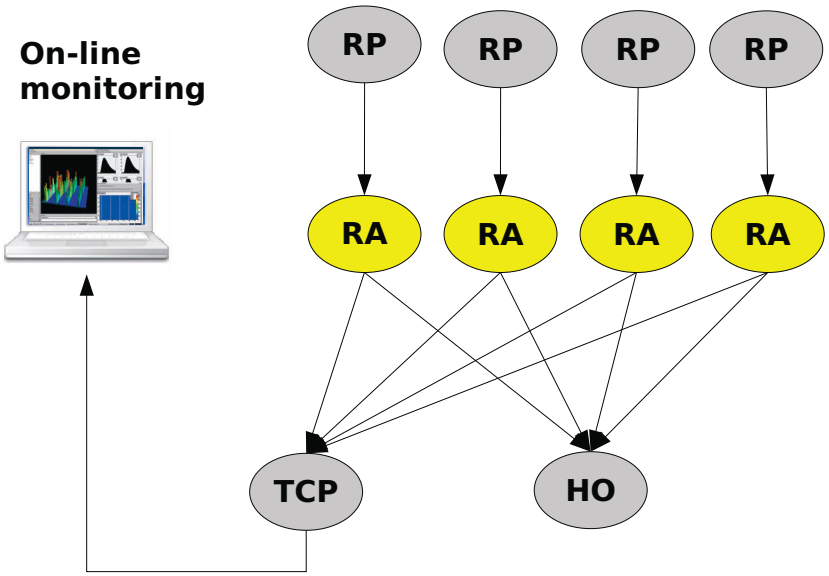


Figure 8.3: Typical HLT configuration used during commissioning at P2. A grey color indicates components that are a part of the HLT framework whereas a yellow color indicates PHOS specific components. RA = *RawAnalyzer*, RP=*RorcPublisher*, HO = *HLTOut*, TCP = *TCPDumpSuscriber*.

number of samples in channels actually containing any data above the zero suppression threshold which for simulated data is set to one ADC count[5]. During real data taking this threshold is typically in the range 2 - 5 ADC counts. For the results shown in Table 8.2 the threshold was set to four ADC counts (for both high and low gain).

For the results shown in Table 8.1 a energy cut of 12 MeV, corresponding to $\sim$2 ADC counts in the high gain channels, was applied to the hits[28] prior to the simulation of the raw data.

The simulation of the raw data assumed a zero suppression threshold of 1 ADC count in both the high and the low gain channel. Furthermore a Gaussian noise of 4 MeV was added prior to the emulated ALTRO digitization.

The average number of samples is therefore given for channels containing data rather that for all channels/events.

[5]During the simulations that were performed so far the electronic noise was assumed to be 0.5 ADC count for all channels.

Event size	440 kB
Occupancy: high gain	19 %
Occupancy: low gain	9 %
Average number of samples: high gain	77
Average number of samples: low gain	37

Table 8.1: Data characteristics for simulated Pb+Pb data assuming $dN/dy \approx 8000$. The number of samples is averaged only over channels containing data. During these simulations it was incorrectly assumed that the sparse readout of the ALTRO chips should be used for Pb+Pb collisions leading to a too low estimate of the event size.

Event size	770 kB
Occupancy: high gain	5 %
Occupancy: low gain	0.1 %
Average number of samples: high gain	105
Average number of samples: low gain	150

Table 8.2: Data characteristics for real zero suppressed cosmic data taken at P2. The number of samples is averaged only over channels containing data. The event size was measured for one module and scaled up to a full PHOS detector with five modules. Sparse readout was not used because it did not work, as a consequence the event size is larger than for simulated Pb+Pb.

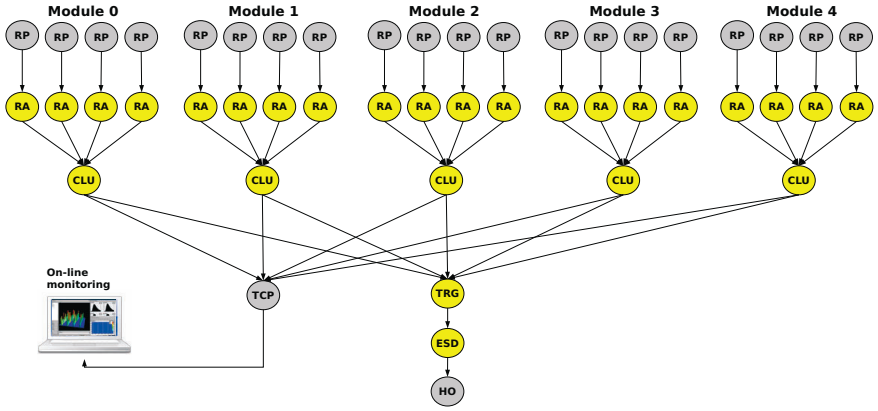


Figure 8.4: Configuration foreseen for the final configuration for the full PHOS detector. A grey color indicates components that is a part of the HLT framework whereas a yellow color indicates PHOS specific components. RA = *RawAnalyzer*, RP=*RorcPublisher*, HO = *HLTOut*, TCP = *TCPDumpSuscriber*, CLU=*Clusterizer*, TRG = *Trigger*. , ESD=*ESD maker*

8.3.2 Data Compression

In the following the discussion the average number of samples for the high gain channels of Table 8.2 will be taken as the most realistic estimate of the data size per channel containing data. The structure currently[6] used for storing amplitude and timing information in the HLT (after processing raw data from individual channels) is shown below in Figure 8.6. The data size of this struct is 24 bytes per channel containing data. The 105 samples of raw data is represented as 10 bit words in the raw data stream giving (10/8)*105 = 131.25 bytes. The data compression factor is 24/131.25 = 5.4. If necessary the data size can be optimized further by representing the energy and TOF by two bytes each and removing the variable representing the data quality. This would give a data size of 6 bytes per channel and a compression factor of ∼20.

8.3.3 Discussion

One could ask why cosmic data was used instead of p+p simulations to estimate event sizes for p+p collisions. The answer is that that the simulations assumes the sparse readout to be activated. It is currently an open question when the sparse readout will be implemented for PHOS, but it will probably not be before the first p+p collisions at LHC. Since the occupancy is low for both cosmic data and p+p collisions this means most

[6]AliRoot trunk, September 2008.

```
  void
  AliHLTPHOSRawAnalyzerPeakFinder::Evaluate(Int_t /*start*/, Int_t length)
  {
1   fDTof = 0;
2   fDAmpl = 0;
3   Int_t tmpLength;
4   if(fTVec == 0 || fAVec == 0){}
5   else
6     {
7       if(length <  fTVecSize){ tmpLength = length;}
8       else{tmpLength = fTVecSize;}
9       for(int i=0; i < tmpLength; i++){fDAmpl += fAVec[i]*fData[i];}
10      for(int i=0; i < tmpLength; i++){fDTof += fTVec[i]*fData[i];}
11      if(fDAmpl > 900){double tmpMax = fUtil->MaxValue(fData, tmpLength);
12      if(tmpMax == 1023){fDAmpl = tmpMax;}}
13      fDTof = fDTof/fDAmpl;
      }
  } //end Evaluate
```

Figure 8.5: The current implementation of the Peak-Finder algorithm in the HLT.

```
struct AliHLTPHOSChannelDataStruct
        {
          UShort_t fChannelID;
          Float_t fEnergy;
          Float_t fTime;
          Short_t fDataQuality;
        };
```

Figure 8.6: Data structure used to hold the reconstructed energy and TOF.

of the data will contain empty ALTRO trailers and noise. The data size for real cosmic data is therefore a better estimate of the real data-size for p+p collisions.

Referring to Table 8.2 it might seem strange at first glance that the average number of samples is higher for the low gain channel than for the high gain channels. The reason is that for cosmic data the signal amplitude for any real external events generally has too low amplitude to be visible also in the low gain channel. The signals seen in the low gain channels are mainly due to noise. Since noise in general is present independently of the trigger it is visible over the whole sampling window, and not just the first ∼100 samples as would be the case for real physical external signals. Consequently the average number of samples will be higher for channels that sees only noise than for channels seeing real signals (provided that the average number of samples is evaluated only from channels containing non zero data).

The figures of merits given in Table 8.1 and 8.2 have some deficiencies and should be used only as a very crude estimate of the event size. The deficiencies are the following: For simulated Pb + Pb data it was later found that due to an error in the simulation of raw data the number of samples was unrealistically high for signals with small amplitudes[7]. On the other hand the data size was underestimated because it was assumed in the simulations that sparse readout is used which is not a realistic scenario for Pb+Pb data (see Section 4.7).

As already described in Section 9.2 the first tests with zero suppressed data revealed that spares readout of the ALTRO chips currently does not work for PHOS. As a consequence the minimum event size is ∼36 kByte [8] per PHOS module and 180 kByte when scaled to five modules, thereby leading to a data volume higher than will be expected for nominal running conditions at the LHC. In the following discussion it will be assumed that the occupancy for cosmic events is comparable to the occupancy in min bias p+p so that the event size obtained for cosmic data can be used as an estimate of the event-size for p+p data.

Referring to the previous section and Table 2.2 an event-size of ∼500 kByte exceeds the anticipated event size of PHOS by a factor 25 for Pb+Pb collisions. This gives a data rate of at least 100 MByte/s for Pb+Pb data at 200Hz and ∼500 MByte/s for p+p data at 1 KHz.

Assuming that available bandwidth for PHOS is proportional to the numbers given in Table 2.2 (0.02x86.5 = MByte 1.73) MByte the bandwidth limitation is then exceeded by a factor ∼800 for Pb+Pb data[9] assuming a bandwidth to storage for ALICE of 1.25

[7]During the simulation of the ADC digitization the time dependent signal was rounded to the closest integer value instead of down to the closest integer value given unrealistic long tails in the simulated digitized signal

[8]In reality the minimum event size is a little bit smaller since the some of the most noisy channels are excluded from the readout

[9]Table 2.2 does not say anything about p+p data, however, if sparse readout is used for p+p data then the event size will be very small.

GByte/s.

To summarize: It is mandatory for PHOS to process and compress the data in the HLT.

8.4 Performance of the Raw Data Analysis Chain

The PHOS raw data analysis chain was bench-marked for performance using VTUNE as a profiling tool as described in Appendix C.

The initial profiling results revealed that the main bottleneck in the performance of the on-line processing was the decoding of the raw data (the decoder). It was concluded that the performance requirements of the HLT could not be met with the decoder provided by ALICE offline analysis code at that moment. This initiated the development of a new decoder that would fulfill the requirements both for ALICE off-line and ALICE-HLT. This development is described further in Appendix D.

New benchmark measurements were performed once the new decoder was ready in July 2007. The benchmarking was done on simulated minimum bias p+p data that was distributed over the PHOS HLT computing nodes on the ALICE-HLT computing cluster at P2. With the new decoder is was found that min bias p+p data could be processed at a rate of $\sim$ 7KHz when only the amplitude and the peak position was evaluated. When clusterization was done in addition the processing achieved an event rate of 2 kHz.

With the new fast decoder it was found, by using VTUNE for profiling, that most of the total CPU time was now spent evaluating the amplitude and the peak-position using the Peak-Finder algorithm, when only the amplitude and the peak position was evaluated. Since it was demonstrated (see Chapter G) that fittings techniques are $\sim$3 order of magnitudes slower, the benchmarking was not performed also for the LMS (The LMS would have been the bottleneck also if the clusterization was done since the speed of the clusterization is in the order of speed of the evaluation of the amplitude and the peak position using the Peak-Finder).

8.5 On-line Monitoring

The monitoring applications (including PHOS) is not a part of the regular processing in the HLT, but receives data from components dedicated to forward data to external applications (for instance the TCPDumpSuscriber). The displaying of the monitoring data is typically done outside the HLT cluster (That is, the monitoring software is not implemented as HLT components).

A typical scenario during debugging and commissioning is to take short runs, typically a few hundred events, examine the data and based on the results of this examination eventually change the configuration of the electronics before taking a new run. This cycle

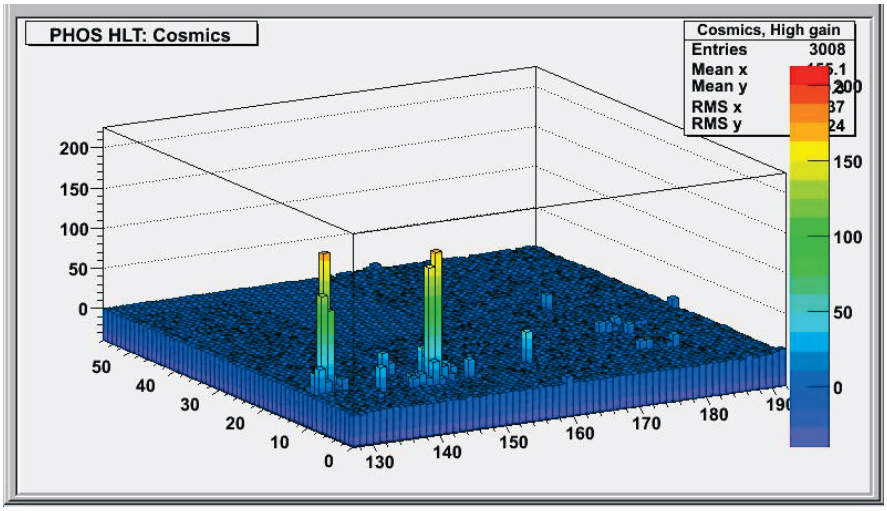


Figure 8.7: Cosmic event taken at the PHOS lab in 2007, possibly a π^0 decaying into $\gamma\gamma$. If this is the case the vertex must be ∼15 m above PHOS, corresponding to roughly the distance between PHOS and the roof of the building where PHOS was situated. That is, the event could be the result of a secondary interaction between a cosmic ray and the roof.

is repeated many times. As an example around 3000 runs was taken solely for debugging purposes during the commissioning of the second PHOS module in the PHOS lab and subsequently at P2 in 2008.

For debugging purposes it is normally preferred to take data without zero suppression so that the data volume is quite large[10]. The examination of the data can be either simple visual inspection of the data or more sophisticated statistical analysis. However, most of the time a simple visual inspection is sufficient after which the data taken has served its purpose and can be discarded. Both disk-space and time can therefore be saved if the examination of the data can be done on-line.

Figure 8.7 and 8.8 illustrates of some of the features of the PHOS on-line display.

8.5.1 Monitoring Data Provided by the PHOS HLT On-Line Display

The PHOS HLT on-line display comprises the following features:

- Event display: Lego and surface plot of the energy deposited in the individual readout channels (see Figure 8.7 and 8.8)

[10]The event size is typically 4MB for one PHOS module

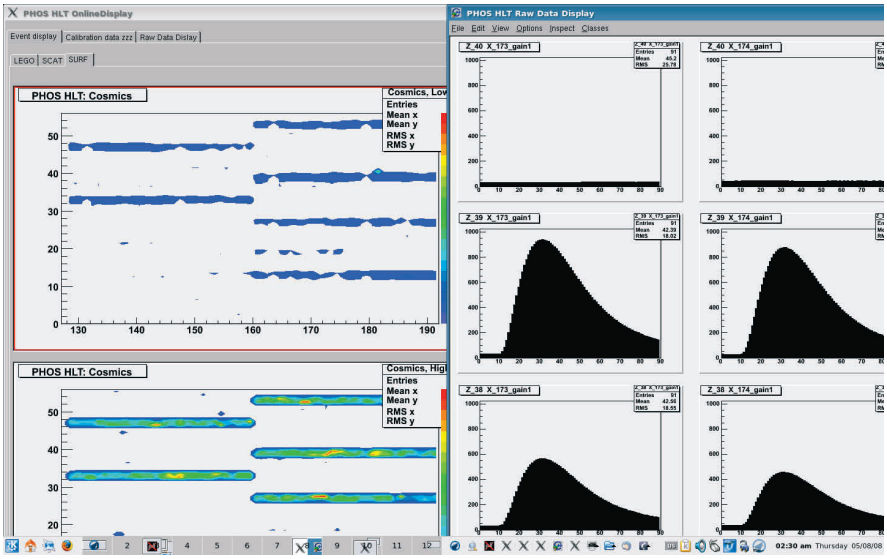


Figure 8.8: Raw data display. By double clicking on the surface plot to the left, the raw data used to evaluate energy and time for that specif crystal and its nearest neighbours (right) are displayed.

- Raw data display: The HLT analysis chain can be configured to attach the raw data to the reconstructed energy. The raw data is displayed by double clicking on the surface event display (see Figure 8.8).
- Dead channel map: Evaluated as the total accumulated energy divided by the number of events (per channel).
- Frequency analysis: On-line FFT (Fast Fourier Transform)

8.5.2 Access Points for the PHOS HLT On-Line Monitoring

In principle the HLT cluster is a closed network, however it can be inconvenient if the detector/HLT expert has to be present *on-site* to debug problems. Monitoring data from the HLT is made available to the outside world by *port-forwarding*. One of the PHOS nodes is set up to forward monitoring data to one of the gateway PCs of the HLT cluster. This port forwarding is based on a server/client paradigm where the server (The HLT gateway machine) forwards monitoring data to a client machine on some external network, but where the server has full control over the data forwarded to the client. From the point of view of security this is completely fine.

As indicated in Figure 2.3 monitoring data from PHOS can be accessed in four ways:

- By an operator logged in to the HLT cluster.
- By an operator logged in to the ACR.
- By an operator with access to the CERN network.
- By an operator with access to the Internet.

The first three access modes were used routinely during the commissioning of PHOS. The fourth access mode was not used so far, but would have been trivial to implement. The only thing that would have been needed was to set up an *shh-tunnel* between the a PC situated inside the CERN and PC on the outside, i.e. on the Internet (see Figure 2.3). The reason why this was not done was that people were mostly present at CERN during the commissioning of PHOS (the monitoring data can then be accessed via the three other modes).

Chapter 9

Commissioning of the First two PHOS Modules

9.1 Commissioning of the First PHOS Module

The first PHOS module was assembled in May - June 2006. After initial testing of the first PHOS module in the PHOS LAB the module was moved to T10 of the CERN PS beam area (refer to Figure 9.1). There were problems experienced during the calibration at CERN PS. These problems are described further in this Section.

9.1.1 Calibration with Electron Beam at CERN PS

The setup for data taking is shown in Figure 9.2. The procedure for calibration with electron beam is described in Section 7.3. During the beam test at CERN PS July - August 2006 calibration data was mostly taken using 2 GeV electron beam. Data was taken also at 1, 2, 3, 4 and 5 GeV in order to measure the energy resolution. Minimum ionizing $\pi^{\pm}$ were studied by placing a copper target in from of the electron beam.

The first step of the calibration is to find the APD DAC settings that equalizes the relative energy output of the detection channels as good as possible as well as making sure that the ADC amplitude is as close as possible to the nominal one i.e 200 ADC counts per GeV. The energy resolution of PHOS is optimized further by applying offline calibration constants. The procedure used to obtain these constants is described in [28].

9.1.1.1 Problems with the Cooling System

The cooling system for the cold volume did not handle the thermal load and the detector was operated between -14°C and -20°C depending on the outside air temperature, instead of the nominal temperature range -25 ± 0.3°C.

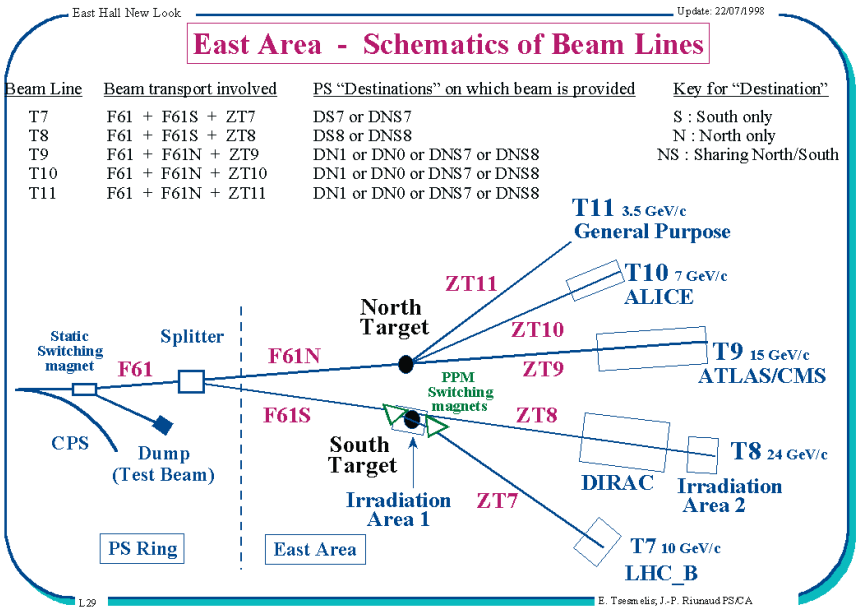


Figure 9.1: CERN PS east area

Figure 9.3 shows the outside air temperature (The zig-zagged line on the top) and the bottom line the temperature inside the cold volume. There were several power cuts at CERN during the summer of 2006. Two of them are indicated in the figure, leading to a malfunction of the cooling system. Except for the two power cuts and an incidence with the cooling system the temperature in the cold volume and the outside air temperature seem to be positively correlated. This means that the cooling system was operated at its limit and that it could not handle the thermal load imposed by cooling the cold volume of the PHOS module down to the nominal $-25°$C.

As a consequence it was decided that a new cooling system was required [1]. The consequence of operating the crystals at $\sim 14°$C instead of the nominal $-25°$C is that the light yield of the crystals drops by a factor 20 %. In addition the amplification of the APDs is reduced when the temperature is increased thereby requiring a higher reverse bias voltage to obtain a nominal APD gain of 50. As a consequence of the latter, several of the APDs would require a HV above 395 Volt i.e outside of the 210 - 395 Volt range of the APD bias control system.

[1] It should be noted here that it was very hot during the summer of 2006 with temperatures reaching as high as 35°C

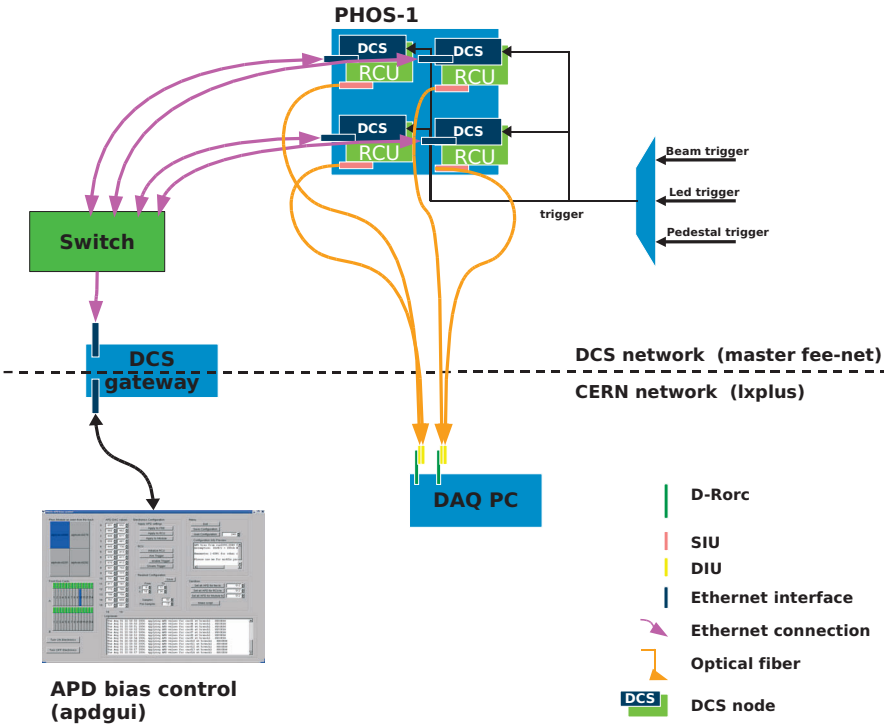


Figure 9.2: Test setup for the electron beam test at CERN PS in 2006.

The L0 trigger efficiency decreases if the detection channels are not well equalized, however since the L0 from PHOS was not used during the electron beam test it was not the major concern. The main issue is that a lot of calibration taken at the wrong temperature has to be extrapolated to -25°C. How accurately this can be done depends on how accurate the temperature dependence of the APD gain and the light yield of the crystals is known. It is outside the scope of this thesis to estimate how accurately the extrapolation can be done but it will nevertheless be stated that it is preferable to calibrate PHOS at the nominal temperature of -25°C degrees. Furthermore the energy resolution deteriorates as a result of the large fluctuations in temperature. Fluctuations in the temperatures can in principle be corrected for, to some degree, during the analysis but is not implemented at present. The reason is probably that with a fluctuation less than ± 0.3°C (as originally expected) this source of error would be negligible.

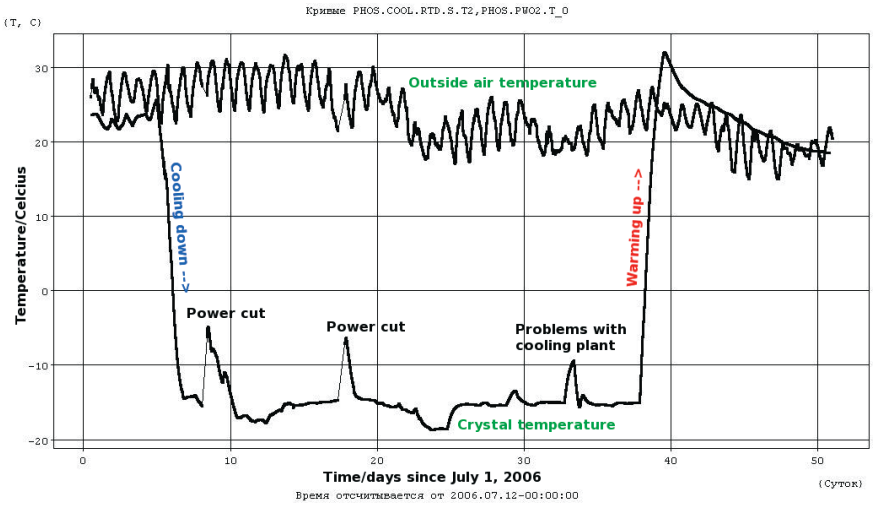


Figure 9.3: zig-zagged line: The outside air temperature. The bottom line: The temperature in the cold volume.The vertical axis shows the temperature in Celsius and the horizontal axis the number of days since July 1, 2006.

9.1.1.2 Problems Configuring the DAC Settings for the APDs

It was found, by reading back the DAC settings from the FEEC, that the DACs on some occasions were not configured for some cards during the configuration phase. The severity of the problem varied from FEEC to FEEC depending on the geometrical location of the card on the GTL bus. The FEECs affected most were those situated on the middle, and at the end of the GTL bus, and for some of those cards the configuration of the DACs could fail 10 - 20 % of the times.

It is clear that taking calibration data without knowing how the detector is configured is meaningless. It was also found that it was mandatory to issue proper feedback to the user about whether or not FEE has been successfully configured. As a temporary solution a read-back/reconfiguration procedure was implemented in DCS. The DAC settings are read back from the board controller each time they are written. The verification and reconfiguration is done on a per FEE basis. If the values read back are not the same as those present in the DCS HV configuration file then the board controller is reconfigured. The reconfiguration is done until the configuration is successful or until a give user defined (typically five) maximum number of configuration attempts have failed. A feedback is given to the user in the apdgui whether or not the configuration of the DACs was successful. If an error has occurred then the DCS software tries to diagnose the problem and displays the appropriate error message to the user. The error messages and their

interpretations are described in Appendix J.6.

9.1.1.3 Problems with Missing or Corrupted Data

It was observed that the data was severely corrupted for some channels in most of the runs. For this reason data was taken mainly without zero suppression. It seemed by visual inspection of the data that corruption occurred by individual bits in the payload changing their value from zero to one and vice versa. If these bit flips occur in the trailer word of the back linked list of ALTRO payloads is broken and the rest of the payload has to be discarded.

Taking data without zero suppression does not cure the problem but provides the possibility to identify it. One easy check that is performed is to check if the payload length indicated by the ALTRO trailer is consistent with the number of pre-samples and samples used during the configuration of the ALTROs.

ALTRO Missing During most of the runs it was observed that data was missing from all 16 channels of individual ALTRO chips. However the chips themselves were not broken as these *missing chips* could vary from run to run. The symptom is that the data is missing for a 2x4 matrix of crystals as indicated in the upper right of Figure 9.4. It was later discovered that the error occurred when writing the Active Channel List (ACL) to the RCU during the configuration phase. The ACL is a 256x16 bit register and each line in this register controls the readout of one ALTRO chip with one bit for each ALTRO channel. If the bit is set (one) the given channel should be read out. When writing the value 0xffff to read out all channels of a chip the registers were sometimes set to zero due to a bug[56] in the RCU firmware. Consequently this chip would not be read out by the RCU.

The temporary solution to the problem was to read back the ACL from the RCU to verify that it is actually set correctly and if not, then rewrite the ACL. This verification/reconfiguration procedure is now done automatically be the PHOS DCS software and there have not been any more problems with missing ALTRO chips.

Broken T-Card: There were problems with broken connections between some of the T-cards and the intermediate PCB feeding the signal through to the warm zone. The symptom is signal missing for a 1x8 matrix of crystals. A typical example is shown in middle bottom and the left edge of Figure 9.4.

FEEC not Working: Some of the FEECs drew a lot of current, probably due to a short circuit, thereby causing the LV power supply to trip. These cards had to be switched off. In addition data was sometimes missing from some FEECs for unknown reasons. A missing FEEC is identified in the data when a 2x16 matrix of crystals are missing in the data read out. An example is shown in Figure 9.4 were 3 FEECs missing.

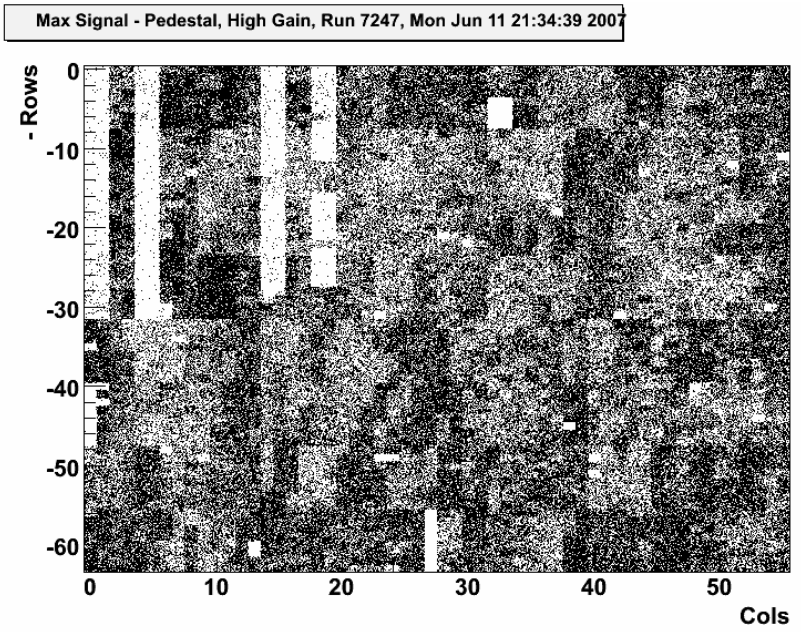
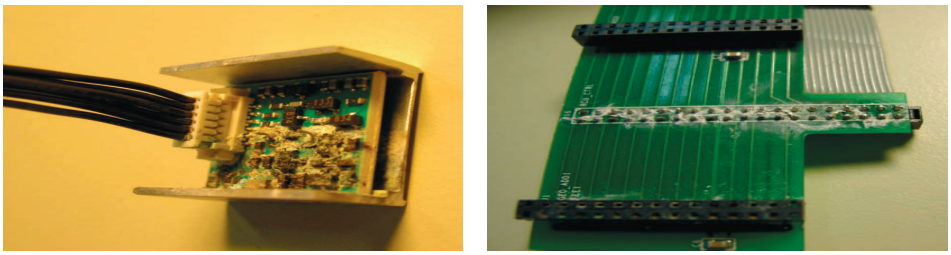


Figure 9.4: The figure shows a dead channels map of PHOS-1 evaluated from run 7247. The data was taken a few months after the electron beam test in 2006 but the problems indicated in the figure are similar to those experienced during the beam test.

Condensation of Water It was suspected that some of the problems with corrupted data and the tripping of the HV were induced by condensation of water on the electronics in the warm zone and CSP + APD in the cold zone. This hypothesis was verified only after the first module was moved to the PHOS lab and opened for inspection (Chapter 9.1.2). Figure 9.5 shows some of the damages caused by the water. Figure 9.5(a) shows a CSP that is destroyed by a short circuit. Figure 9.5(b) shows corrosion caused induced by a short circuit.



(a) Corrosion on CSP.

(b) Corrosion on a connector on the GTL bus.

Figure 9.5: Corrosion caused by condensation of water.

9.1.1.4 Results

Despite all the problems enough data was accumulated to obtain APD calibration for approximately 2/3 of the module using the edge of the ADC spectrum as explained in Section 7.3.

It was found that the gains of the detection channels could be equalized within $\sim 4\%$ by tuning the APD configuration. The energy resolution was found to be $\sim$3.3% at 2 GeV. The energy resolution measured at 1, 3, 4 and 5 GeV is shown in Figure 4.12.

9.1.2 Calibration with Cosmic Muons

After the electron beam test the module was moved to the PHOS lab and data taking was continued using cosmic rays as probes, using both the TRU and two vertical layers of coincidence scintillator for the trigger. The setup is shown in Figure 9.6.

The PHOS module was flushed with nitrogen in an attempt to solve the problem with condensation of water experienced during the electron beam test.

For the HLT the primary objective was to test the PHOS HLT software in an on-line environment similar to what would later be used in ALICE. If MIP signals were observed in data selected by the HLT on-line it would be a proof that the HLT is able to select useful data and compress the data volume by discarding the rest. The object for the HLT

was thus just to observe a MIP peak in individual channels, but not necessarily to use this data for calibration.

For calibration purposes the data was scrutinized by more accurate during off-line analysis.

The PHOS module was orientated with the crystals in zenith angle. Cosmic data was taken both with coincident scintillation trigger and the TRU. The data taking using the TRU as a trigger was abandoned at an early stage firstly due to the problems described in Section 9.1.1. Secondly, a coincident scintillation trigger gives a better selection of MIP candidates since triggers are issued only for particles coming from the vertical direction. For calibration purposes one is mainly interested in particles that deposit all of their ionization energies in a single crystal. The selection of data was performed in five stages, in the order given below.

- Coincidence scintillator trigger
- Zero suppression in the HLT
- Discarding of noisy data
- Clusterizing using a dedicated MIP clusterizing algorithm
- Isolation cuts on the MIP clusters

The experimental setup is shown in Figure 9.6. The two vertical scintillation triggers were placed with a separation of around 1.2m on the top[2] of the PHOS module. One FEP was borrowed from the HLT cluster and placed in the PHOS lab. For practical purposes the FEP PC was connected to the CERN GPN network[3]. This has the advantage that monitoring data from the HLT is directly available from any other machine on the CERN network without further action. In addition the DAQ computer which is also connected the CERN GPN network will then be visible from the FEP node so that the data could be written back to the DAQ PC using a simple NFS mount of the DAQ hard-drive to the FEP node[4]. This way of writing the data back to DAQ was more than sufficient for the data taking with cosmics since the event rate and the event sizes were relatively small. For large data volumes it would have been too slow.

[2]It would have been more efficient to have placed one scintillator on top and the other one below the PHOS module in order to select particles that goes trough the whole detector. The reason why this was not done was purely practical due to lack of space below the module when rotated into vertical position.

[3]In the ALICE experiment the HLT nodes are only accessible from within the HLT cluster

[4]This solution is highly *non standard* and created a lot of turmoil in ALICE since the official mechanism of writing the output data to the HLT over the HLT out PCI card and DDL is bypassed.

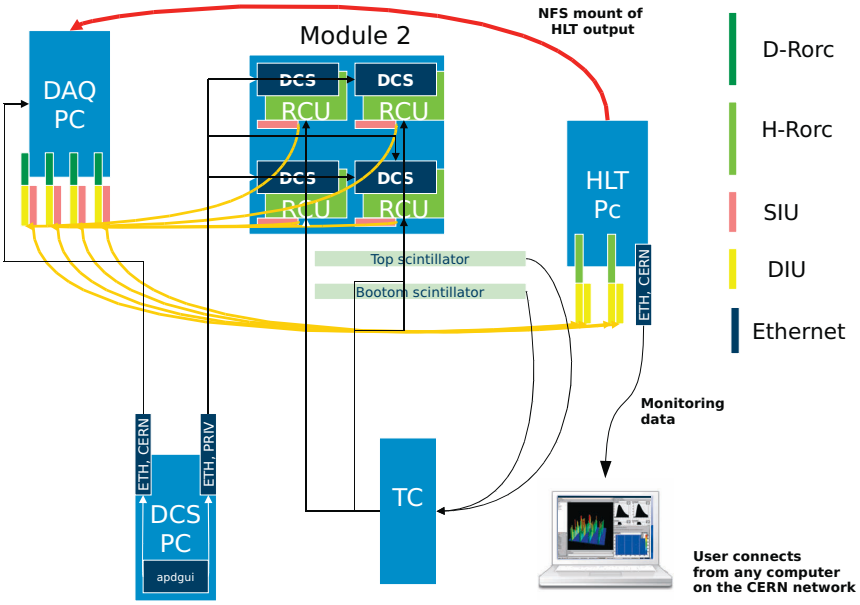


Figure 9.6: Test setup for calibration with cosmic muons in the PHOS LAB in 2006.

Cosmic Trigger The first selection of useful MIP candidates is performed by separating the two scintillator triggers in the vertical direction by a distance that is large compared to the sensitive trigger area in order to ensure that triggers are issued only for MIPS coming from a small solid angle around zenith angle.

9.1.2.1 Trigger and HLT Analysis Chain

Due to problems with corrupted data zero suppression could not be used. As a consequence the signal to background ratio for MIPs was quite bad since a MIP given signal in a single channel required the readout of the full PHOS module (7168 channels) with a predefined number of samples for for each channel. In order to reduce the amount of data to handle during further offline analysis the data was compressed in the HLT by writing to storage only data above a given threshold. The per channel baseline was evaluated with dedicated baseline runs. During the on-line analysis in the HLT the baseline values were subtracted from the time dependent signal of each individual channel. Then a user defined global threshold was varied and applied pragmatically[5]. It was found that a threshold of 2 ADC counts gave a data compression rate of about 50, and acceptable trigger bias.

Discarding of Noisy Data Examining the data after zero suppression in the HLT revealed a high contamination of noise. Therefore signal shape analysis was applied prior to the clusterization and only signals consistent with a gamma-2 shape were used.

Fitting techniques were found too slow and were therefore not used. Instead a pragmatic approach was implemented noting that for a bounded function with well defined integral in the range $\pm\infty$, and with a signal shape independent of the amplitude, the integral of the signal divided with the amplitude must be a constant defining a quality factor "Q"

$$Q = \frac{\int_{-\inf}^{\inf} f(x)dx}{f(x)_{max}}$$

Or for a digitized signal

$$Q = \frac{\sum_{n=i_{star}}^{i_{end}} x(n)}{x_{max}}$$

For a digitized signal the Q value will depend on how many samples are used and in which range.

To first approximation the signal is assumed to be a gamma-2 function and the Q value can be evaluated analytically. Small deviations from an ideal gamma-2 signal shape can be accounted for by allowing signals within a user defined a range of Q values to pass

[5]A more sophisticated scheme using individual threshold for the channels based on dedicate noise measurements was planned, but not implemented.

the filter. The range of Q values used during analysis was found by trial and error and was adjusted so that all signals with a gamma-2 or a gamma-2 like shape, were accepted through the filter. The Q value was typically chosen in the range 27 - 33.

A *bad* Q value thereby guarantees that the signal is not physical, but noise. On the contrary a *good* Q value does not guarantee that the signal is not noise so that noise events can still pass through the filter. No systematic studies were done on the signal to noise ratio after this filtering, however visual inspection of around 100 events revealed no suspicious data.

Clusterization Since the MIPs do not develop electromagnetic showers and since the total deposited energy is small, the size of the clusters rarely extend beyond a 3x3 matrix of crystals, in addition clusters from different MIPs rarely overlap. These two assumptions above were utilized in the implementation of a dedicated clusterization algorithm that is simpler and less sophisticated (and fatser) than the general on-line clusterizer algorithm, but also faster.

Isolation Cuts The MIP clusterizer gives as output MIP clusters from particles that can traverse more than one crystal. It should also be taken into account that not all the cosmic shower particles are MIPs.

The single crystal MIPS can be singled out by applying an isolation cut on the energy deposited in the central crystal of the 3x3 shower, i.e. by requiring a certain fraction of the total energy of the cluster to be contained in the central crystal. If one wants to single out MIPs that traverse just one crystal, it is tempting to set this cut to 100 % however, noise just above threshold in neighbouring channels as well as cross talk can lead to a substantial amount of the total energy being deposited in neighbouring crystals. The data was therefore analyzed with different isolation cuts starting with 0% and increased in steps of 10% up to 100%. Figure shows 9.7 the resulting energy distribution after applying these cuts.

9.1.2.2 Problems

Condensation of Water It was found that the flushing of the warm volume with nitrogen did not solve problem with condensation of water. It was believed that PHOS module was not air tight enough, leaving pockets of air inside the module. It was concluded that PHOS could not be operated at -25°C with the current design. As a consequence it was decided that a new air tight design of the module cage was needed.

In addition the damages caused by short circuits and corrosion lead to decision to focusing on getting the second PHOS module ready for the first LHC beam and that this module should be operated at room temperature until the new air tight design was ready.

TRU Triggering on Noise Examination of data taken using the TRU revealed that most of the time (97 %) it was triggering on noise. The trigger threshold was carefully fine tuned for each individual 2x2 analogue sum. For reasons currently not understood the noise levels rose drastically when starting DAQ and consequently the DAQ bandwidth was swamped with noise.

9.1.2.3 Results

HLT Figure show an energy distribution for 3x3 clusters for a single crystal (the central crystal of the 3x3 being always the same) using data selected by the HLT. The data was taken using coincidence scintillator trigger. The distribution shown in the figure is typical for crystals covered by the trigger. Figure 9.7(a) - 9.7(k) shows the energy distributions when different selection cuts on the energy deposited in the central crystal of the 3x3 sum is applied. Figure 9.7(a) show the energy distribution when no cut is applied and Figure 9.7(k) the distribution when all the energy is deposited in the central crystal. The intermediate figures 9.7(b) - 9.7(j) show the energy distributions resulting from increasingly larger selection cut in steps of 10%.

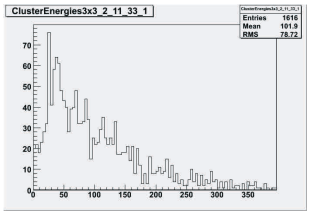
The clearly visible MIP peaks in Figure 9.7 show that the mission was accomplished for HLT.

Calibration Data The cosmic data was scrutinized further during off-line analysis. Results are reported in [51]. Enough data was analyzed to provide calibration for ∼900 crystals. It was found that the spread of the MIP peaks was 18%. Since the APD settings used to obtain 4% calibration for 2 GeV electron beam were used one would expect a similar result for MIPs. In addition it was found that the correlation between the deposited energy for MIPs and 2 GeV electrons was much lower than expected.

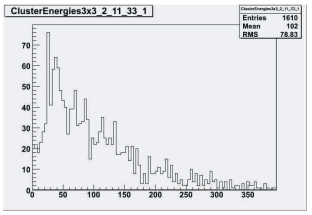
The reason for the discrepancies is not completely understood but the combination of the larger spread of the MIP peaks and the small correlation points towards a non-linearity in the absorption of energy by the $PbWO_4$ which varies considerably from crystal to crystal.

9.2 Commissioning of the second PHOS module

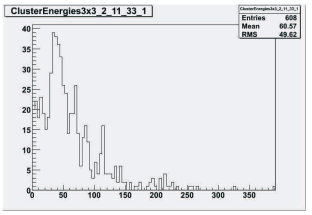
The numerous problems encountered with the first PHOS module led to the decision that this module should be dismounted to investigate the problem with condensation and other problems further. It was decided that the first priority should be given to the second PHOS module to get it ready in time for the first LHC run. There were sever problems also with the second PHOS module, both with corrupted data (Section 9.2.2) and with noise (Section 9.2.3).



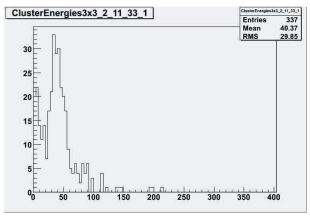
(a) No cut



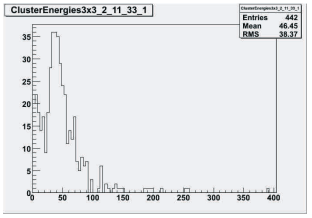
(b) 10% of total cluster energy deposited in central crystal



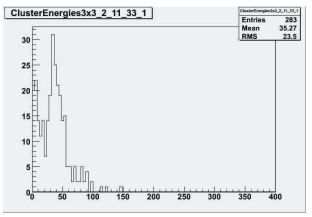
(c) 20% of total cluster energy deposited in central crystal



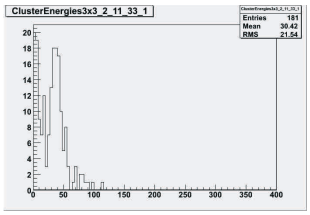
(d) 30% of total cluster energy deposited in central crystal



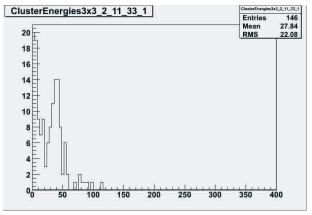
(e) 40% of total cluster energy deposited in central crystal



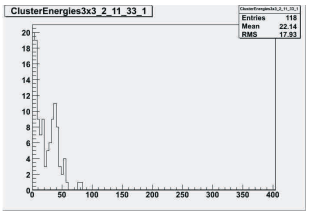
(f) 50% of total cluster energy deposited in central crystal



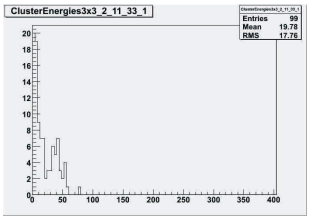
(g) 60% of total cluster energy deposited in central crystal



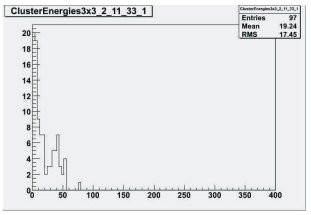
(h) 70% of total cluster energy deposited in central crystal



(i) 80% of total cluster energy deposited in central crystal



(j) 90% of total cluster energy deposited in central crystal



(k) all energy deposited in central crystal

Figure 9.7: Energy distribution of 3x3 clustered, centered on a single crystal for different isolation cuts.

Gain	Channels requiring more than 400 V/%
50	37.33
30	1.64
20	0.53

Table 9.1: The percentage of APDS requiring more than 400 V in order to obtain gain 50, 30 and 20.

Experiences from commissioning of the first PHOS module led to several changes listed below.

- The mechanical designed was made air tight to allow flushing with over-pressurized nitrogen
- It was decided to operate the module at room temperature (18°C) until the problems with condensation are fully understood and a solution is found.
- Improved monitoring software
- Software for automatic testing of electronics

The consequence of operating PHOS at room temperature is that many of the APDs require more than the maximum 400 V reverse bias voltage to obtain the nominal gain of 50. Table 9.1 shows the percentage of APDs requiring a reverse bias voltage of more than 400 V in order to obtain a gain of 50, 30 and 20 respectively.

9.2.1 Software Tools

A substantial collection of tools were developed during the commissioning of the second PHOS module. The developments were driven by day to day experience and problems at P2. Most of the tools were developed from scratch. The collection of tools denoted *phos_ana* was collected in a single repository. The software is available at http://github.com/perthi/. The package depends on ROOT for histogramming and Fourier analysis, but is independent of AliRoot. A description of some of the tools is provided below. The name in parenthesis is the actual directory name in the repository.

PHOS Utilities (phosutils) Contains utilities used by the other tools: 1) Several histogram classes which inherit from one and two dimensional histograms in ROOT. The histogram classes have axes, contour levels.. etc tailor made for PHOS. 2) A tool for parsing of command line arguments. 3) An event-loop that takes care of the handling of the raw data. The other tools typically inherit from this raw data loop. 4) A facility to easily write images on any format to file (uses ROOT). Many of the figures in this thesis are produced by this facility and used unmodified.

Fourier Analysis (fourier_ana) Estimation of the frequency spectrum, the power spectral density and the phase (developed for investigation of noise problems, described in detail in Appendix E.2.2). Figure 9.18 and 9.17(b) are examples of figures produced by this tool.

Analysis of Pedestal Data (pedestal_ana): This is the most frequently used tool. The minimum output of the tool is five histograms, a two dimensional and a one dimensional histogram for both high and low gain showing the noise in the PHOS module and one histogram showing the HG/LG noise ratio. An example of such histograms is shown in for instance Figure 9.19.

The noise is estimated with the technique described in Appendix E. Several command line arguments can be given when executing the executable *pedestalana*. The only mandatory one is the run number. A list of directories specified in the file *config.txt* are searched for data from run with the given run number. If data from this run is found the pedestal analysis is started, if not the execution is aborted. The file *config.txt* must be present in the directory where the executable is started. To start the analysis of lets say run 51248 one would type

```
./pedestalana  -run  51248
```

The run number will appear in the title of all histograms. The switch *-fourier* triggers the frequency and phase analysis.

If the switch -psd is used in addition the power spectral density will be estimated using the Fourier transform of the auto-correlation as an estimate (refer to Appendix E).

The switch *-thr <threshold>* produces an active channels list that can be used during the configuration of the electronics from DCS. The hardware addresses of channels with noise levels less than <threshold> are written to the list. This feature is used in order to mask out noisy channels from the readout.

In addition there are several other options. The full list of options is written to the screen by using the *-help* switch.

Only global features of the module are plotted as default. For the Fourier analysis the frequency spectra are averaged over the whole module for high and low gain separately. Information about individual channels is available by clicking with the mouse cursor on the two dimensional noise plot. Each bin, that is each *(z, x)* coordinate, is associated with a linked list, of arbitrary length, of channel specific histograms. At least one histogram is always associated to each coordinate, namely the pedestal histogram used to measure the noise of the channel. If also the Fourier and the power spectrum analysis is performed then per channel specific frequency spectra will be available in addition. Figure 9.8 shows an example.

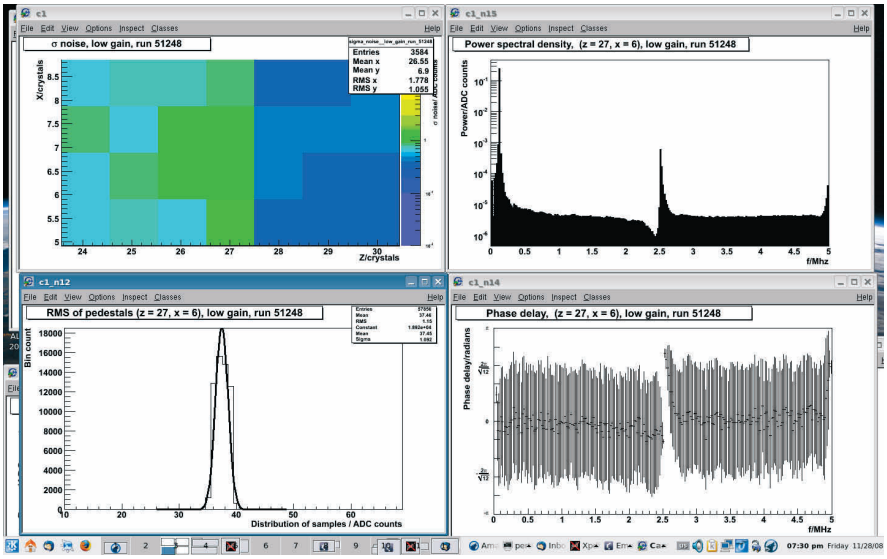


Figure 9.8: Some of the outputs from the pedestal analysis. The user can zoom in on regions on the two dimensional noise plot to the upper left by clicking and dragging with the mouse on the z and x axis. By clicking with the mouse on individual bins on the noise plots several channel specific histograms and plots are display. Some of them are shown to the right and at the bottom.

Correlation Analysis (noise_covariance): The program measures the correlation coefficient between the time dependent signal of single reference channels, and all the other channels. Figure 9.16 shows an example of the output of this program.

Rawdatadisplay (rawdatadisplay): Displays raw data for a 1x1, 3x3, 5x5, 7x7 or 9x9 matrix surrounding an an arbitrary channel specified on the command line. All the raw data figures presented in this thesis are created with this tool.

Occupancy Measurement (occupancy) Calculates the occupancy and the average event size. The program uses the same event-loop as all the other tools.

Estimation DAC Settings from Hamamatsu Data (HV_ana): The directory contains two tools. One is a root macro *HV_ana.C* which takes a list of voltage settings from Hamamatsu and generates a DAC configuration file that can be used by the *apdgui* to configure the voltage settings of the APDs. In addition there is a tool to measure the correlation between the DAC settings and the noise.

9.2.2 Investigation of Problems with Corrupted Data

The initial test of the readout after the assembly of the second PHOS module revealed that the data was severely corrupted. In the beginning it was the exception rather than the rule that it was even possible to take data with DAQ, and when it was possible to take data it was commonly the case that it was not possible to read it.

9.2.2.1 Basic Testing of the Digital Communication

The first step to understand the problems with corrupted data was to perform single transaction tests between the RCU and the FEE. Dedicated software was developed that made it possible to measure the transaction rate error for single transaction between the RCU and the BC, the RCU and the ALTRO registers and between the DCS card and the RCU (Section 5.6). The transactions between the RCU and the BC were tested both over the SC and the ALTRO bus[6].

9.2.2.2 Testing the Digital Communication with the ALTRO Pattern Playback

In order to debug the problem with corrupted data independently from problems with noise the electronics were configured for *pattern playback mode*. In this mode of operation the physical signal sampled by the ALTRO ADC is ignored by the ALTRO data assembler

[6]The ALTRO chip can only be accessed over the ALTRO bus

and upon the arrival of a trigger only the pedestal memory is copied to the output-buffer of the ALTRO. This gives the possibility to test and validate the digital communication between the FEE and the RCU as well as between DCS and the RCU.

For a given run all the channels were configured with the same pattern however this global pattern was typically varied from run to run. Visual inspection of data taken in pattern playback mode revealed that something was obviously wrong with the readout for several of the channels. To summarize, the following errors were observed.

1. Data completely missing for some individual channels

2. Read-back pattern is different from the input pattern

   Pattern has spikes that vary in position and magnitude from event to event, and from channel to channel

   Pattern has missing values (zeroes), however the pattern is the same within a given readout partition

Systematic Investigation After the first tests with ALTRO pattern read-back it was clear that a more systematic study of the problem was needed in order to solve the it. Figure 9.12 gives a schematically overview of the methodology. The FEE including the ALTRO chips is configured by DCS prior to data taking. In order to automatize and simplify the testing the same DCS configuration file is used both by the DCS software that configures the electronics and the program used to analyze the results. At the time this software was developed it was found useful to be able to do this testing on-line. The pattern validation was therefore implemented as an HLT component that can be run either on-line or off-line. The debugging consisted simply of comparing the pattern read back from the electronics with the pattern used to configure the electronics and to count the number of patterns observed in the data both for the module as a whole and for each individual readout channel. If there are no digital errors anywhere during configuration or read-back then exactly one pattern should be observed in all the channels of the whole module and this pattern should be the same as the pattern used to configure the electronics. If more than one pattern is observed, measurements on the electronics and more low level functional tests, such as those described in Section 9.2.2.1, are initiated. The debugging cycle is shown schematically in Figure 9.13. From the point of view of the digital communication the following errors might occur.

1. Errors when writing data to the RCU

2. Errors when transferring the data from the RCU to the ALTRO chip

3. Errors during read-back of the data

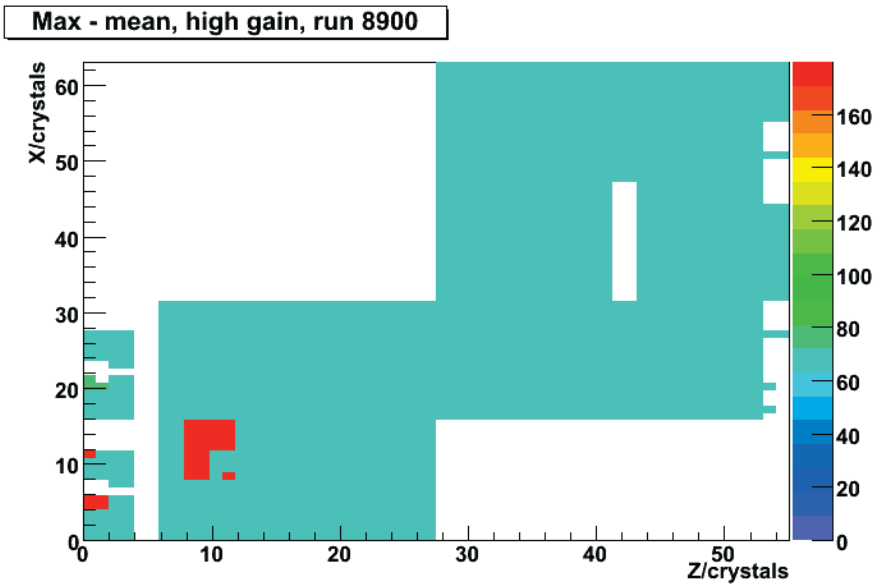
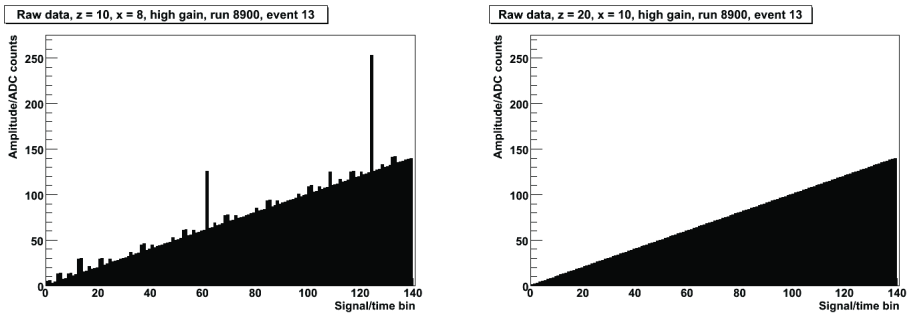


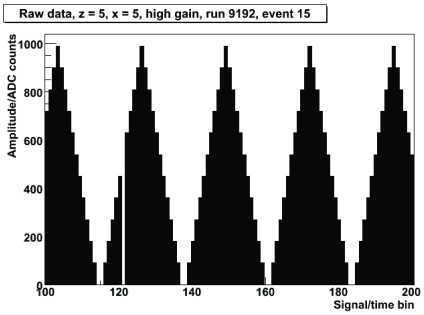
Figure 9.9: Run 8900. Color coded surface plot of maximum signal values. The red areas are due to spikes. A turquoise color indicates signals with expected maximum values. The white areas indicates that there is no data. For the upper left quadrant data is missing because the DAQ was configured not to read out this partition (there were only three available working RCUs at the time). For the remaining white areas data was lost either because of corrupted data or because of malfunction of some of the FEECs. The red areas is due to *spikes* in the time dependent signal as indicated in Figure 9.10(a).



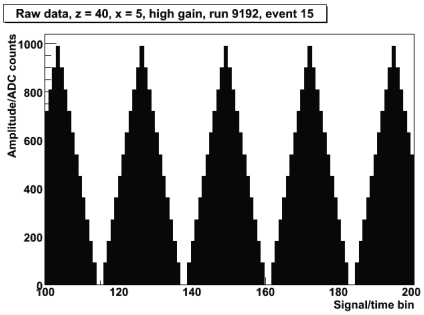
(a) Typical corrupted channel

(b) Typical good channel

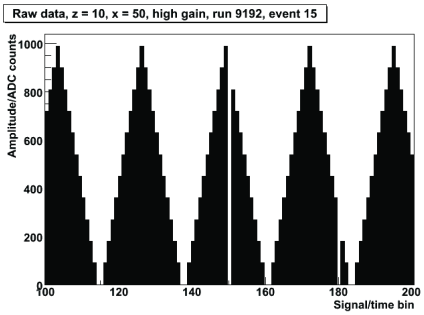
Figure 9.10: run8900: Ramp function loaded into the pedestal memory of the ALTRO and read back upon the arrival of a trigger.



(a) Pattern seen in data coming from RCU0



(b) Pattern seen in data coming from RCU1



(c) Pattern seen in data coming from RCU3

Figure 9.11: Run 9192: 3 out of 4 partitions were configured for readout. Partition 0 and 3 have errors in the pattern, the error is the same for all events and all channels and for the given partition. The pattern is as expected for partition 1. No data was taken for partition 2. The channels were configured with a sawtooth pattern with 300 samples. Only time bins in the range between 100 and 200 are displayed. Outside this range the pattern is as expected.

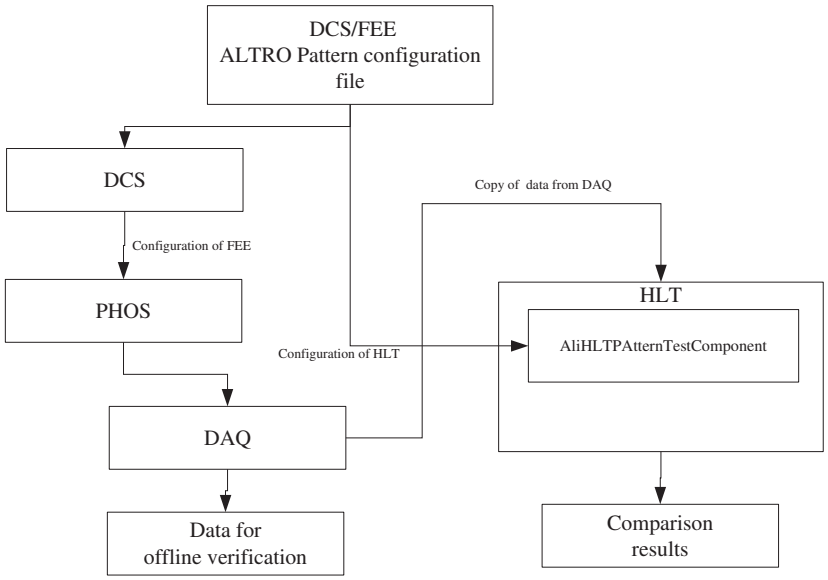


Figure 9.12: Debugging the HLT using the *pattern test* component.

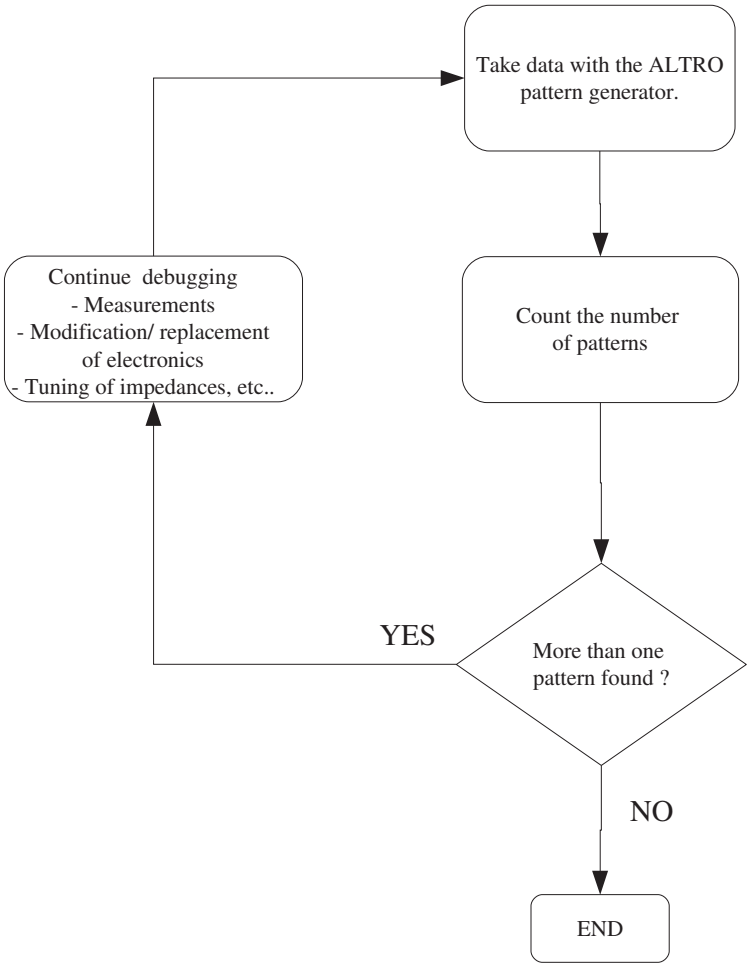


Figure 9.13: Debugging cycle when solving the problem with corrupted data.

Error 1 manifests itself as a pattern that differs from partition (RCU) to partition, but that is the same for most of the channels within a single partition. Figure 9.11 shows patterns that are typical for this kind of error.

Errors of type 2 are associated with the writing of the pattern to individual channels (in broadcast mode) from the RCU to the ALTRO. For such errors one expects a pattern that is different from the input pattern and also different from the other channels on the same readout partition, however if there is no problem during the readout this pattern will be the same from event to event for an individual channel. This kind of error can be distinguished from error of type 1 by counting the number of patterns observed within a single partition.

Errors of type 3 typically manifests as patterns that change from event to event for an individual channel, however it can also happen that there are errors in the digital communication during read-back that always affects the same data lines. Therefore a pattern that changes from event to event for a single channel can be unambiguously assigned to an error of type 3 On the other hand, if the pattern is different from the expected one, but is constant from event to event, then it can be due to an error of either type 2 or type 3.

9.2.2.3 Evaluation of two Different Board Controllers

Two different BCs were developed, one at Wuhan, China and another one at the University of Bergen, Norway. The two board controllers were benchmarked against each other measuring the performance for low level transaction tests (See Section 9.2.2.1). The results from the transaction tests using the two board controllers are presented in Appendix I.

The two board controllers were also benchmarked against each-other during regular data taking using the ALTRO *pattern playback mode*. The two chosen observables were 1) The number of events that had to be discarded altogether due to corrupted data, 2) The number of patterns observed in the data for the events that could be correctly decoded. There was assumed to be no inconsistency in the data if there was exactly one pattern observed in all the readout channels (a ramp pattern as shown in Figure 9.10(b) was used). If more than one pattern is observed then this is a proof that some of the data is corrupted.

The results from the pattern playback test using both board controllers are shown in Table 9.2. The results given in Appendix I and Table 9.2 show that the Bergen BC is

BC	version	Discarded data blocks	Number of patterns
Wuhan	3.0	8.7 %	12
Bergen	3.4	0.9 %	1

Table 9.2: Discarded data blocks and observed patterns using the Wuhan and the Bergen BC respectively.

more robust against corrupted data and more reliable during data taking. In addition [57]

it has been reported that for the Wuhan BC there were timing violations on the BC side during transactions between the RCU and the BC, possibly causing the data to be more corrupted. It was therefore decided that the BC developed at Bergen should be used for PHOS.

9.2.2.4 Summary and Conclusion

The observed problem with corrupted data was found to have several causes listed below.

- Broken GTL bus drivers on some of the RCUs
- Bad impedance matching between the RCU and the GTL back-plane
- Improper grounding of the GTL back-plane
- A bug in the RCU firmware, leading to, on some occasions, a failure to write values to the RCU registers (including the pedestal memory interface)

The problems with broken GTL drivers on the RCUs could only be fixed by replacing the corresponding RCU. Bad impedance matching between the RCU and and the GTL bus was fixed by measuring the impedance of the individual GTL bus and matching the termination resistors on the GTL buses accordingly. Regarding improper grounding of the GTL bus it seems that the errors originating from this problem can also to some extent be cured by fine tuning the impedance matching, however, the current solution has been found unsatisfactory and a new solution is currently under testing.

9.2.3 Investigation of Problems with Noise

The noise measurements presented in this section were performed on signals after the ADC quantization with the ALTRO chip.

In addition analogue noise measurements were performed. Some of the results are presented in Appendix F.

After the problems with corrupted data were solved, measurements taken in the PHOS lab revealed noise levels acceptable for physics. The noise levels were of the order of the ADC quantization error for most channels. Appendix F shows the results for a series of noise measurements taken in the PHOS lab just before installation in ALICE.

New noise measurements were taken after the second PHOS module was moved to the ALICE experiment. Initially these measurements revealed noise levels comparable to those observed in the lab. However, after several weeks of operation the noise suddenly increased. A typical example of a time dependent signal under noisy conditions is shown in Figure 9.14. The noise seemed to consist mainly of a sinus component with an modulated amplitude.

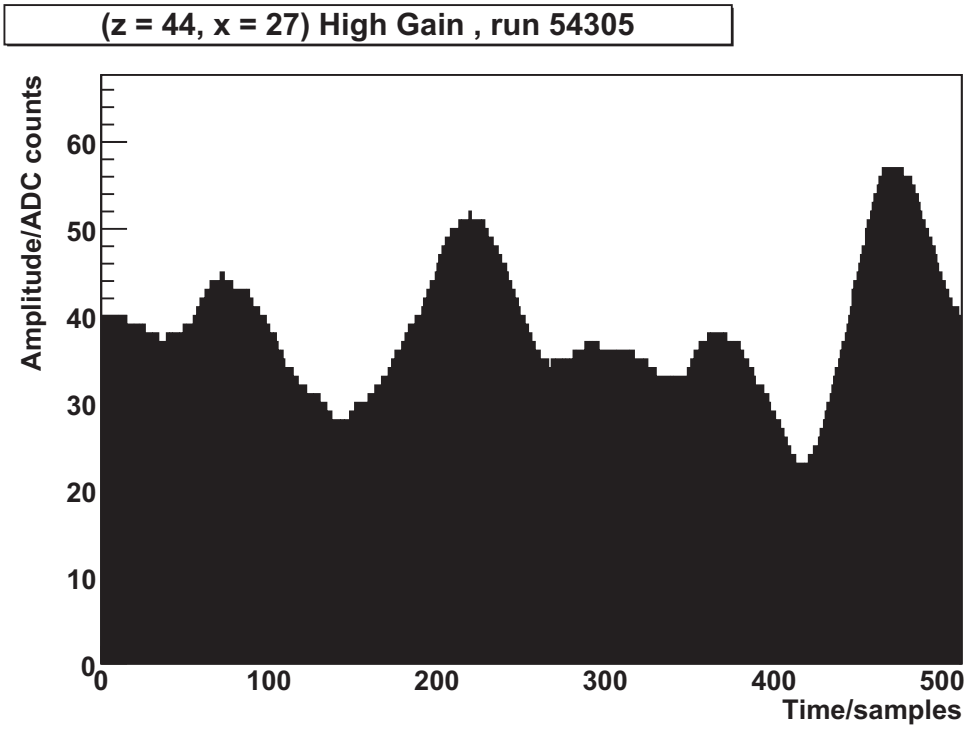
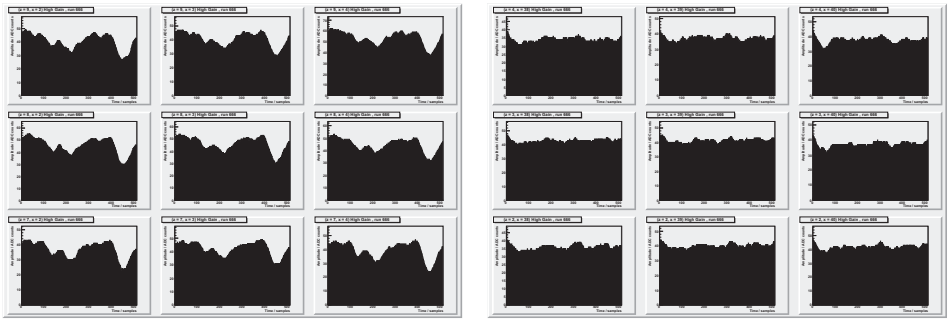


Figure 9.14: Time dependent signal for a typical noisy channel.

In addition the following interesting features were observed: Firstly; the magnitude of the noise seemed to be branch dependent. Secondly; the time dependent signal seemed to be almost identical in all channels within a single branch. Figure 9.15 shows two



(a) branch 0 (b) branch 1

Figure 9.15: Typical time dependent signal under noisy conditions. The figure shows 3x3 matrix of signals for the same event but for two different branches.

3x3 matrices of time dependent signals for a typical event taken under noisy conditions whereas Figure 9.15(a) shows time dependent signals from branch zero whereas Figure 9.15(b) show the time dependent signal in branch four for the same event. The figures shown are typical. As seen in the figure the signals from different channels on the same branch are highly correlated whereas it is not obvious, at least not by visual inspection, that the signals are correlated between channels from different branches.

9.2.3.1 Measurements of the Correlation Coefficient Between Different Channels

In order to assess more quantitatively how the signals are correlated, the correlation coefficient was measured as following: An arbitrary[7] channel on one of the branches was selected as reference and the correlation coefficients between the reference channels and all the other channels were evaluated. The correlation coefficient was estimated both for single events and as an average over several event.

Figure 9.16 shows the correlation coefficient averaged over 211 events. The correlation coefficient obtained for individual events looks similar. The reference channel is z=10, x 36. As seen in the figure the correlation is almost one between the reference channel and channels sitting on the same branch. In addition it is interesting to note that the correlation coefficient for channels sitting on other branches than the reference channel is

[7]The channel was not selected completely arbitrarily but was chosen among some of the most noisy channels

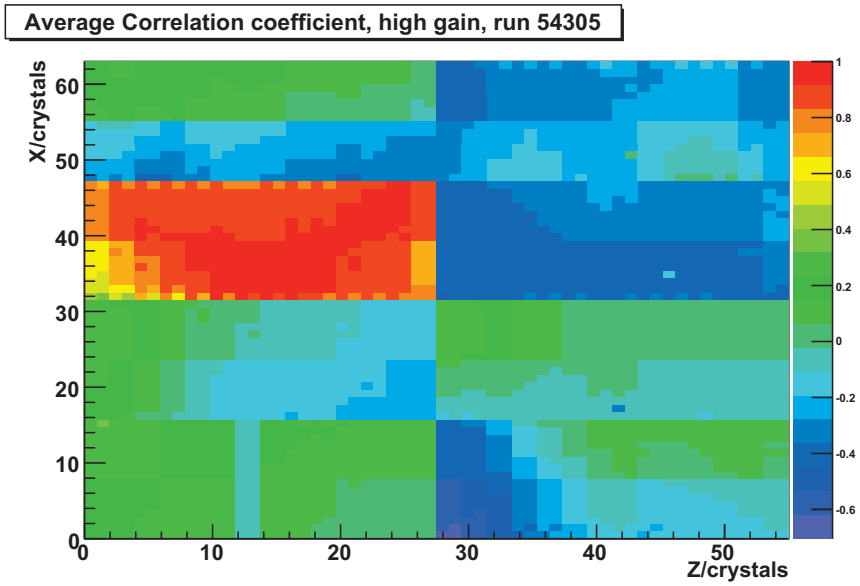
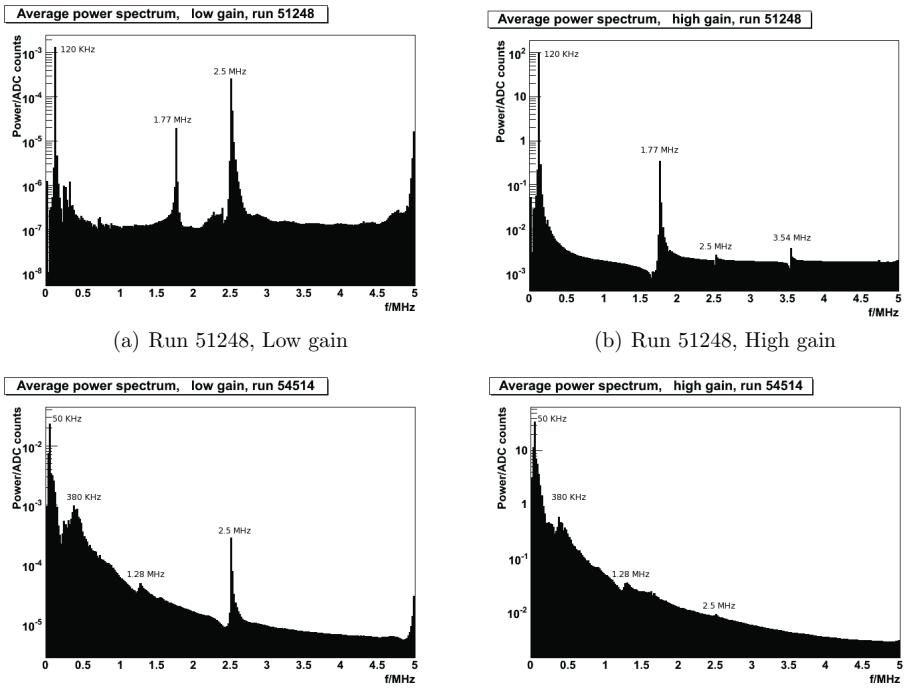


Figure 9.16: Correlation coefficient averaged over 211 events of time dependent signals between a reference channel (z = 10,x = 36) and all the other channels.

significantly different from zero when averaged over 211 events. One plausible explanation is that there is a fault in the HV power supply that affects all eight HV output channels, creating a sinus like noise which for a given branch is in the same phase for all channels, while the phase is different from branch to branch so that the signals are indirectly correlated between the branches through this sinus noise.

9.2.3.2 Frequency and Phase Distribution of the Noise

Power spectrum Referring to Figure 9.17(a) and 9.17(b) taken under high noise conditions and Figure 9.17(c) and 9.17(d) taken under normal noise conditions it is clear that the excessive noise is associated mainly with a single frequency component at around ∼120 KHz. The power spectra in these and all the other figure presented in this Section were evaluated as described in Appendix E. In addition the Figure 9.17(c) and 9.17(d) reveal



(a) Run 51248, Low gain

(b) Run 51248, High gain

(c) Run 54514, Low gain

(d) Run 54514, High gain

Figure 9.17: Run 51248 and 54514: Power spectral density for high and low gain respectively

frequency components that are not visible in figure in Figure 9.17(a) and 9.17(b) namely

at ∼50 KHz, ∼380 KHz, ∼1.28 MHz. Furthermore there is and ∼2.5 MHz component visible in all plots.

The frequency resolution of the discrete Fourier transform is given by $\Delta_f = \frac{f_s/2}{N_{samples}}$. For the measurements presented in this section the sampling rate was $f_s = 10$ MHz (The sampling frequency used by the ALTRO) and N = 512 samples giving a frequency resolution of ∼10 kHz. The estimated peak frequencies therefore have an uncertainty of ±10 kHz.

The origin of the ∼50 KHz peak is unknown. The peak at ∼380 KHz was observed also during the commissioning of the first PHOS module. Since this noise was measured on the Wiener LV power supply it was assumed that the Wiener was the source, however we shall see in Section 9.2.3.7 that this was probably an incorrect conclusion. The frequency component at 1.28 MHz has as origin a single *hot* channel that affects some of the neighbouring channels via cross talk and that for unknown reasons seems to travel between 1.9MHz and 1.2 MHz and with an amplitude high enough that it is visible when all channels are averaged over the whole module.

Phase Distribution During the commissioning of the first PHOS module it was observed, when testing the TRU, that the noise increased during data taking. One of the theories about the origin of the noise was therefore that it was induced by our own data taking via the ADC sapling clock.

Noise components affected by the ADC sampling clock, or the system wide clock signal should also be correlated with the clock signal. Consequently the noise should be in a well defined phase relative to the clock.

In order to investigate this theory further the phase delay relative to the sampling clock was measured on an event to event basis, then averaged both globally for the whole PHOS module and for each individual channel.

Figure 9.18 shows a typical phase distribution as a function of frequency. The figure was obtained by analyzing ∼ 200 events and the variance of the phase was evaluated directly from the data. The error bars indicate the standard deviation measured from the data.

For frequency components which are independent of the data taking the phase relative to the ADC sampling clock should be uniformly distributed in the range $[-\pi, \cdots, \pi]$.

The definition of the variance is

$$Var(X) = \sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2 f(x) dx \tag{9.1}$$

where X is a random variable with probability density function $f(x)$ and the expectation value $\mu = E\{X\}$. For a random phase delay θ uniformly distributed in the range $[-\pi, \cdots, \pi]$ we have $E\{\theta\} = 0$ and $f(x) = 1/2\pi$ giving a variance of

$$Var(\theta) = \int_{-\pi}^{\pi} \frac{\theta^2}{2\pi^2} d\theta = \frac{\pi^2}{3}. \tag{9.2}$$

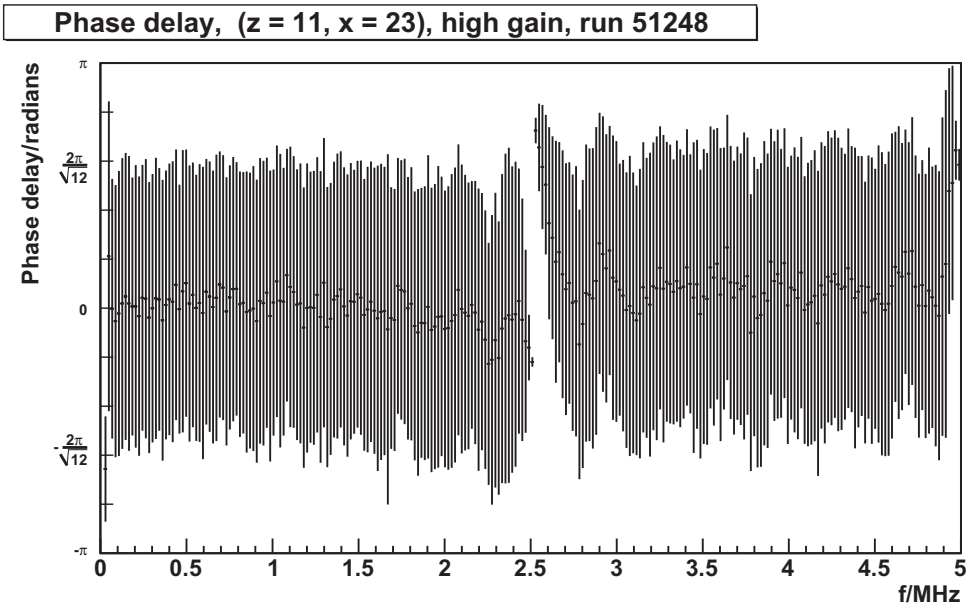


Figure 9.18: Typical phase distribution for a typical noisy channel.

The standard deviation is

$$\sigma = \sqrt{Var(\theta)} = \frac{2\pi}{\sqrt{12}}. \tag{9.3}$$

We write $2\pi/\sqrt{12}$ instead of $\pi/\sqrt{3}$ because of the resemblance with the standard deviation $(1/\sqrt{12})$ of the uniform distribution in the interval $[0, \cdots, 1]$. As seen from Figure 9.18 the phase delay is consistent with random phase for all frequency except the component at 2.5 MHz.

The 2.5 MHz ALTRO Buffering Noise The only frequency component which is obviously correlated with the ADC sampling clock is the one at 2.5 MHz. This frequency cannot be due to an external physical signal since the -20 db cutoff frequency of the shaper is ∼1 MHz.

The explanation is the following: The ALTRO samples at 10 MHz with a dynamic range of 10 bit whereas the data bus is 40 bit wide. The ALTRO assembles four 10 bit samples and writes them to the internal multi event buffer at every fourth clock cycle i.e with a frequency of 2.5 MHz.

When data is taken without zero suppression and with the same number of pre-samples then this buffering is done simultaneously for all 7168 readout channels of the PHOS module. This simultaneous buffering draws enough current to distort the digital ground, and thereby the reference voltage of the ALTRO ADCs, enough to produce a 2.5 MHz noise component that is correlated with the sampling clock.

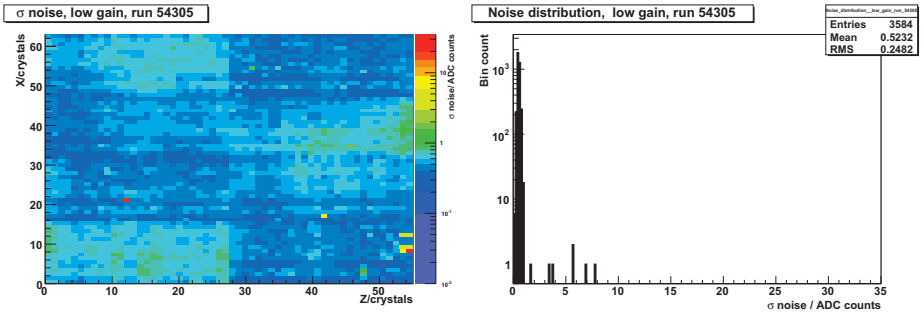
It is assumed that the 2.5 MHz noise will be significantly reduced when zero suppression is used since the buffering them will be asynchronous.

For the TPC, the ALTRO buffering noise was significantly reduced for non-zero-suppressed data by varying the number of pre-samples between channels on the same ALTRO chip [58]. For PHOS the maximum number of pre-samples is needed in order to recover the full signal. This is therefore not a viable option for PHOS.

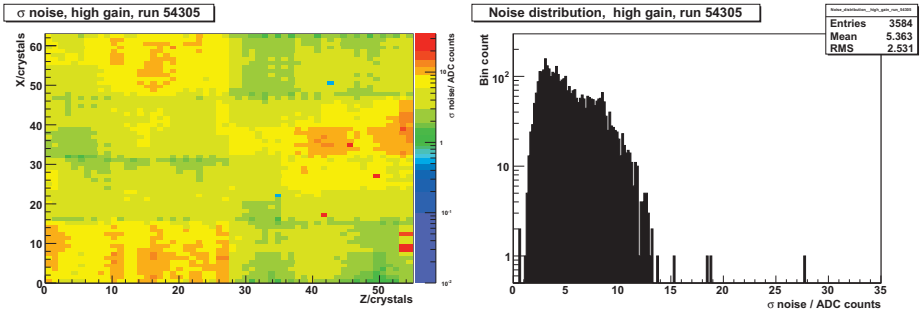
9.2.3.3 Noise Before and After Modifying the Grounding

It was speculated that the cause of the noise problem was due to an unfortunate grounding scheme creating a ground loop between the high and low voltage power-supplies. It turned out later on that nobody really knew how the power supplies were grounded initially, but they where probably grounded via the power outlets. Nevertheless, the grounding was modified by connecting the ground of the power supplies to the cage of the module. Figure 9.19 shows the noise just before the modification. As seen in the figure the noise is very high with noise levels > 5 ADC counts for most of the high gain channels. In addition there is a branch like structure of the noise. This structure will be discussed further in Section 9.2.3.8.

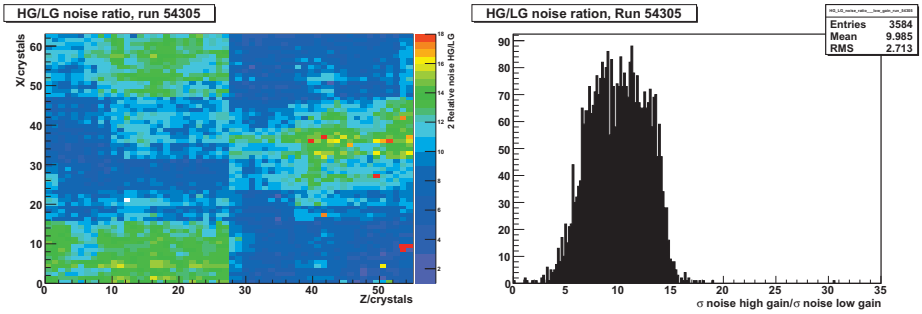
The noise levels after modification of the grounding shown in Figure 9.20 reveal a much better noise condition with acceptable noise levels of ∼1 ADC counts for most high gain



(a) Low gain



(b) High gain



(c) High gain/low gain noise ratio

Figure 9.19: Noise just before modifying grounding.

channels and less than 0.5 ADC counts for most low gain channels. It would be tempting to conclude that the cause of the noise problem was due to bad grounding however one should keep in mind that the modification involves a power cycle of PHOS since the power supplies had to be switched off during the modification. Such a power cycle have on some occasions made the noise problem disappear.

The ground connection between the module cage HV + LV power supplies was removed in order to test if the disappearance of the noise was due to the modified grounding. If so, then the noise should reappear. Unfortunately, as shown in Figure 9.21 the noise level was more or less unchanged after going back to the initial grounding. It is therefore not possible to draw any conclusion about whether or not the noise problem is related to the grounding of the power supplies.

9.2.3.4 Noise Dependence on the APD Configuration

The noise was measured for three different APD gains (20, 30 and 50) while keeping the output voltage constant at 395 V for all eight output channels. The results are shown in Figure 9.22. As seen in the figure the noise is basically the same for all three gains.

9.2.3.5 Noise as a Function of the ISEG Output Voltage

Figure 9.23(a) to 9.23(f) show the noise when the ISEG HV voltage is varied in the range 100 V to 395 V while the APD gain is kept constant at 50. As seen in the figure it is clear that the noise levels depend on the output-voltage of the ISEG. Similar measurements were performed a few weeks later but the observations could not be reproduced.

9.2.3.6 Influence of Other Sub-Detectors and the L3 Magnet

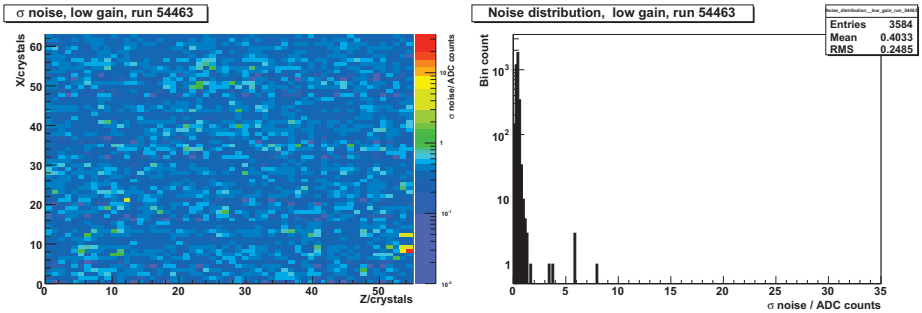
It was speculated if the noise observed in PHOS was pick-up noise produced by other detectors or the L3 magnet. A series of noise measurements were performed with TPC, TRD, TOF and the L3 magnet ON vs OFF, taking data/idle (most permutations were tried). To summarize the result: There were no observed correlation between the operation of other detectors or the L3 magnet and the noise observed in PHOS.

9.2.3.7 Analogue Noise Measurements

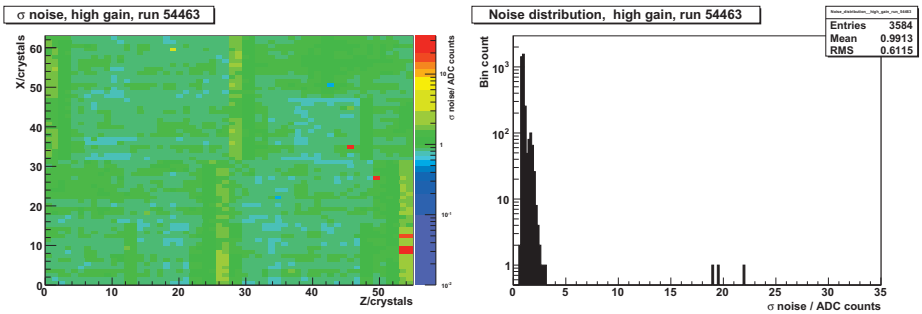
The noise was measured directly on the grounding cables, HV/LV cables and the flat-band cable connecting the LED crate with the LEDs as indicated in Figure F.6. The results are shown in Figure F.7 to F.13.

To summarize the observations (taken fro [59]).

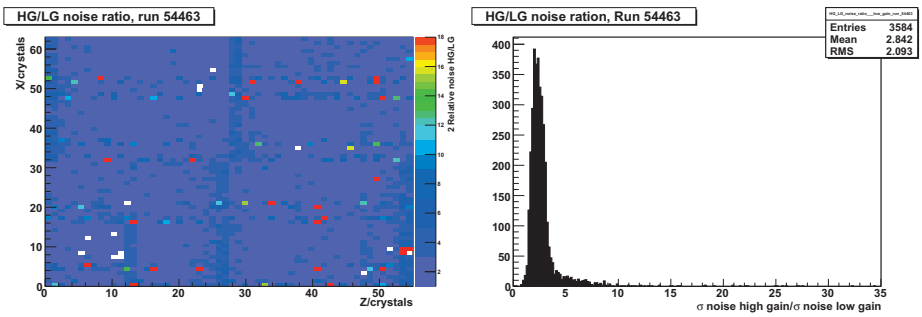
- The noise measurements in the PIT show two main noise components, one at ∼375 kHz and one at ∼650 kHz.



(a) Low gain

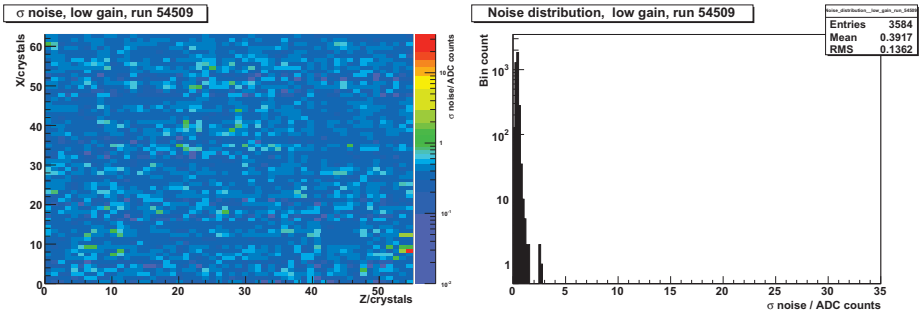


(b) High gain

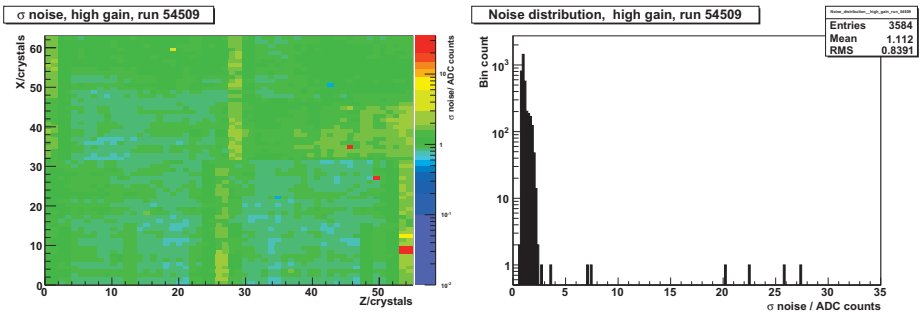


(c) High gain/low gain noise ratio

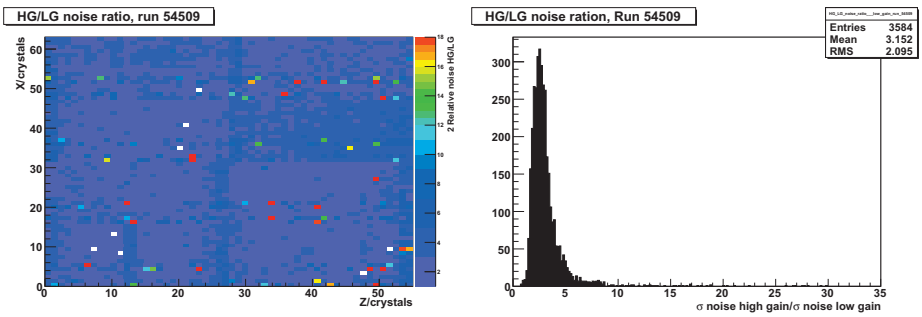
Figure 9.20: Noise with modified grounding.



(a) Low gain

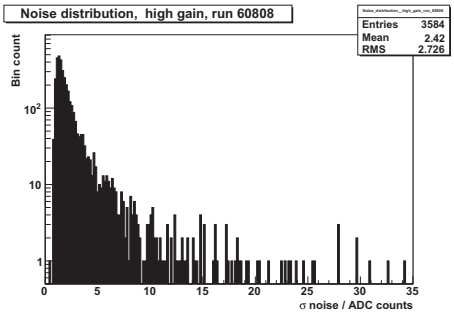


(b) High gain

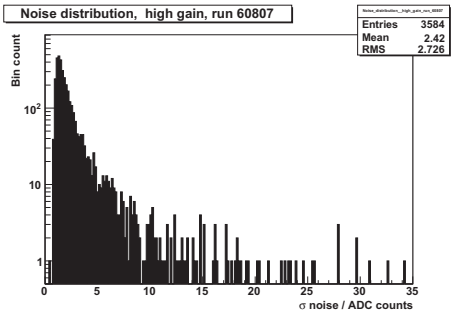


(c) High gain/low gain noise ratio

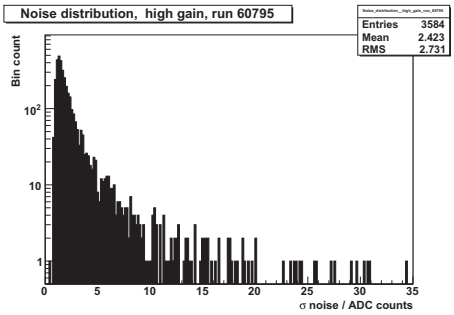
Figure 9.21: Noise after going back to original grounding.



(a) APD gain 20, ISEG Voltage = 395. (data taken at P2 25/09/2008)

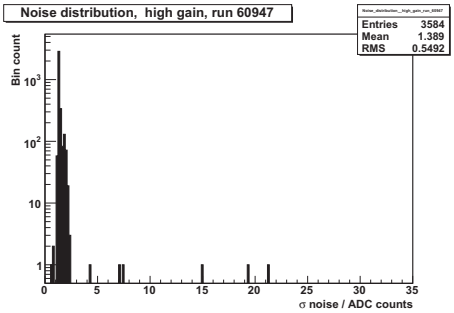


(b) APD gain 30, ISEG Voltage = 395. (data taken at P2 25/09/2008)

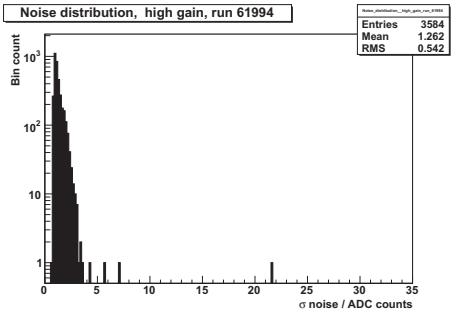


(c) APD gain 50, ISEG Voltage = 395. (data taken at P2 25/09/2008)

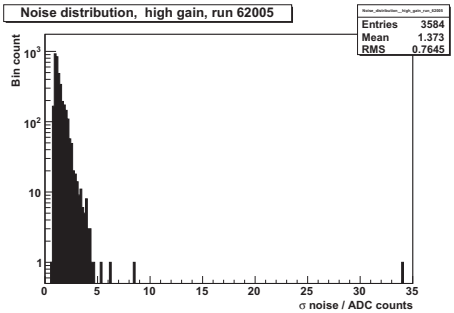
Figure 9.22: The noise dependence of APD DAC settings at ISEG voltage 395.



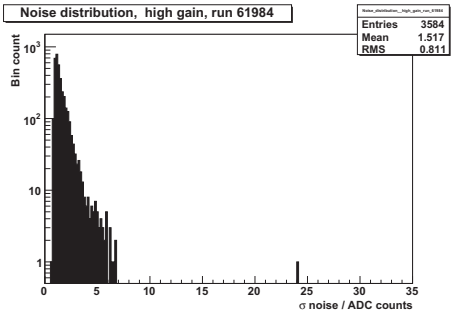
(a) APD gain 50, ISEG Voltage = 100. (data taken at P2 27/9/2008)



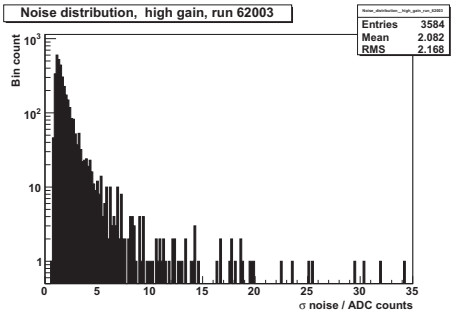
(b) APD gain 50, ISEG Voltage = 370. (data taken at P2 6/10/2008)



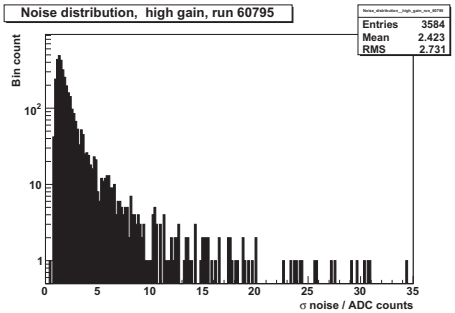
(c) APD gain 50, ISEG Voltage = 375. (data taken at P2 6/10/2008)



(d) APD gain 50, ISEG Voltage = 380. (data taken at P2 6/10/2008)



(e) APD gain 50, ISEG Voltage = 390. (data taken at P2 6/10/2008)



(f) APD gain 50, ISEG Voltage = 395. (data taken at P2 25/9/2008)

Figure 9.23: The noise dependence on the ISEG output HV.

- The ∼375 kHz noise is due to an un-shielded CANBUS cable to the temperature sensors in the cold volume running alongside un-shielded HV cables.
- 650 kHz noise component is measured on the flat cable between the LED crate and the LEDs.
- The 120 kHz noise which was seen in the data of many of the runs in June - November 2008 was not seen during the noise measurements in the PIT, neither was it seen in data taken at the same time.

The 650 kHz noise is seen only on flat cable for the LED system.

The ∼375 kHz noise component originates from the unshielded CANBUS cable and is picked up by the HV cables, and the digital and analogue ground ground of the Wiener power supply. This noise is also consistent with the ∼380 kHz noise seen in the data, taking into account the ∼10 kHz frequency resolution of the Fourier analysis.

9.2.3.8 Discussion

As seen in Figure 9.17(c) and 9.17(d) the ∼380 kHz pick-up noise has a relatively small contribution to the power spectrum compared to other frequency maxima. The reason is that this frequency is attenuated by ∼15 db by the shaper (see Figure 4.6).

The 120 kHz peak on the contrary is close to the maximum frequency response of the shaper (see Figure 4.6) and therefore benefits from the full signal amplification[8].

There are several observations pointing towards the ISEG HV power supply as the source of the 120 kHz noise.

1) By comparing Figure 4.11 with some of the noise measurements performed under noisy conditions, for instance Figure 9.19 or Figure F.2, it is clearly seen that the noise has a branch like structure. The only voltage supply which has dedicated, per branch, output channels is the high voltage from the ISEG (as seen from Figure 4.11. The LV is distributed on the module level).

2) The high gain to low gain ratio is ∼16 for the most noisy channels (see Figure F.2). This means that the noise originates before the shaper in the signal chain.

3) The time dependent signal for channels on the same branch has a correlation coefficient of almost exactly one. Between different branches the correlation coefficient is non-zero. This means that all the branches sees the same 120 kHz noise, but in different phase (Section 9.2.3.1). If the noise was due to an external source it is hard to understand why it should be in different phases in different branches.

4) The noise increases with increasing output voltages of the ISEG while keeping the HV DAC configuration constant at M=50 (Figure 9.23). On the contrary the noise was not affected significantly when the HV was kept constant at 395 V and the HV DAC configuration was varied (Figure 9.22).

[8] Assuming the 120 kHz noise enters the readout chain before the shaper.

Furthermore, it was speculated if the observed noise signal was due to actual light hitting the APDs, for instance due to some malfunction of the LED system, however if this would have been the case one would expect that the magnitude of the noise to increase for increasing DAC HV settings which is clearly not the case (see Figure 9.22). The branch like structure of the noise also makes this scenario unlikely since the LED system does not operate on the branch level.

Finally, it has been speculated that the combined system of FEE and power supplies + grounding etc.in effect acts as an oscillator, and that since the terminal capacitance of the APDs decrease with increasing reverse bias voltage, then the resonance frequency depends on the bias voltage of the APDs. That is, if there is an external 120 kHz noise source then this noise starts to affect the signal when the resonance frequency reaches 120 kHz. This scenario is however not very likely for the following reason: The terminal capacitance is affected[9] by the reverse bias voltage only when the voltage is less than the depletion voltage. Above the depletion voltage the capacitance is independent of the bias voltage. The 210 minimum voltage exceeds the depletion voltage of the APDs [60]. Regulating the voltage in the range 210 - 400 V should thereforE not affect the capacitance of the APDs.

Summary & conclusion The observed problems with noise (in the data) is most likely due to a fault in the HV power-supply.

[9]The terminal APD capacitance decreases for increasing reverse bias voltages

Chapter 10

Summary

The main topic of this thesis has been the commissioning of the PHOS detector and the integration into the ALICE High Level Trigger.

10.1 DCS

In order to commission a detector one first has to control it. This control is provided by the Detector Control System. A detector control system for the front-end electronics, suitable for the operation of the full PHOS detector, was therefore developed and used during the commissioning of the first two PHOS modules. This system controls the bias voltages of the APDs and the readout configuration of the electronics.

10.2 HLT

During the commissioning one wants to to monitor the detector response. It is an advantage if the detector response can be monitored on-line. Since PHOS is a detector that participates in the HLT it is natural that the on-line monitoring is provided by the HLT. Performing the on-line monitoring in the HLT also served the purpose of commission the HLT software since the on-line event reconstruction has to work in order for the monitoring of the reconstructed data to work.

For physics data taking the HLT has to process data at the full event rate. The first step in the event reconstruction is to evaluate the amplitude and the peak position from the digitized signals. If standard LMS fitting techniques had been used for this purpose, this would have been the bottleneck in the on-line event reconstruction, making the on-line reconstruction in the HLT too slow to cope with the full event rate in min. bias p+p and central Pb+Pb collisions.

A different approach to the estimation of the amplitude and peak position of a digitized signal, the Peak-Finder algorithm, was tested and implemented. The output of the

algorithm converges to optimum estimates 2 - 4 orders of magnitude faster than fitting techniques, thereby satisfying the performance requirements of the HLT. The Peak-Finder and the LMS algorithms are equivalent with respect to accuracy.

10.3 Commissioning of the First two PHOS Modules

Two PHOS modules were assembled and tested at CERN. One module was installed and commissioned in ALICE and was ready for the anticipated first LHC run in September 2008, however, due to an accident with one of the LHC magnets the first collisions will be produced at earliest in June 2009.

There were problems with corrupted data with both modules which were subsequently solved. For the second PHOS module there were in addition problems with noise.

Commissioning of the First PHOS Module The first PHOS module was tested with electron beam at CERN PS in 2006. The energy resolution was found to be satisfactory, however, there were problems with condensation of water on the electronics due to insufficient insulation between the warm and the cold zone.

The first PHOS module was moved back to the PHOS lab after the beam test and the calibration was continued using cosmic muons. The cosmic calibration also provided the first successful test of the PHOS HLT software in a realistic on-line environment.

The attempt to solve the problem with condensation of water by flushing the module with nitrogen failed. The reason was that the current design was not air tight enough leaving pockets of air inside the module. It was concluded that a new air tight mechanical design was needed.

The first module was completely disassembled in order to investigate the damages and to fully understand the problem with condensation of water.

Commissioning of the Second PHOS Module The focus was shifted towards getting the second PHOS module ready for the first collisions. It was decided that this module should be operated at room temperature until a new air tight design was ready. The second PHOS module was installed in ALICE and fully integrated into all on-line systems. The PHOS HLT software was successfully tested. Excellent performance and a data compression with a factor $\sim$15 was demonstrated.

There were severe problems with noise after the installation of the second PHOS module in ALICE. The problems are currently not completely understood, but there are indications that the source of the noise is the HV power supply.

10.4 Status and Outlook

New mechanical design fulfilling the air tightness requirement is ready at CERN. The second PHOS module was moved back to the PHOS lab for modification. Two additional modules are currently being assembled at CERN. Three modules with air tight design operated at $-25°$C will be installed in ALICE before the first p+p collisions at LHC in 2009.

The second PHOS module was fully integrated with all on-line system in ALICE, including the HLT. For the PHOS HLT, the event reconstruction and data compression were successfully tested., however, the PHOS HLT was not yet used as a trigger. The development of the trigger software and the integration with other detectors is currently ongoing.

Furthermore, the Peak-Finder algorithm can be optimized by solving the optimum Peak-Finder equations as outlined in Appendix G.1.5.1.

Part III

Appendix

Appendix A

List of Publications

- Hille, Per Thomas; ALICE Collaboration; *The Photon Spectrometer of the ALICE Experiment* Preceedings for PANIC 2008; Preprint submitted to Elsevier
- ALICE Collaboration; *ALICE Technical Paper 1.* CERN October 2007
- Alme, J; Campagnolo, R; Djuvsland, O; Fehlker, D; Gutierrez, CG; Helstrup, H; Hille, Per Thomas; Haland, O; Muller, H; Munkejord, M; Musa, L; Karlsson, ADO; Pimenta, R; Richter, M; Rossebo, A; Roed, K; Rohrich, D; Skaali, Toralf Bernhard; Stangeland, A; Ullaland, K. *Radiation-tolerant, SRAM-FPGA based trigger and readout electronics for the ALICE experiment.* IEEE Transactions on Nuclear Science 2008; Volum 55. s. 76-83
- Alme, Johan; Helstrup, Haavard; Hille, Per Thomas; Muller, Hans; Musa, Luciano; Oltean, Alexandra; Pimenta, Rui; Richter, Mathias; Røed, Ketil; Roehrich, Dieter; Skaali, Toralf Bernhard; Ullaland, Kjetil. *Radiation-tolerant, SRAM-FPGA Based Trigger and Readout Electronics for the ALICE Experiment.* 15th IEEE NPSS Real Time Conference 2007; 2007-04-12 - 2007-05-04
- Bruna, E.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration. *Open charm reconstruction in ALICE:* $D^+ \rightarrow K^-\pi^+\pi^+$. Quark Matter 2006; 2006-11-14 - 2006-11-20
- Conesa del Valle, Z.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Electroweak boson detection in the ALICE muon spectrometer.* European Physical Journal C 2007 ;Volum 49. s. 149-154
- Dainese, A.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *ALICE perspectives for the study of charm and beauty energy loss at the LHC.* European Physical Journal C 2007 ;Volum 49. s. 135-141
- Guernane, R.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration. *Heavy flavour production in ALICE.* Symposium on Hadron Collider Physics; 2006-05-22 - 2006-05-26
- Heinz, M.T.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration. *Reconstructing Bottom mesons using displaced vertices from semi-leptonic decays.* 23rd Winter Workshop on Nuclear Dynamics; 2007-02-11 - 2007-02-18

- Jorgensen, C.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *First physics with ALICE.* Acta Physica Polonica B 2007 ;Volum 38. s. 1001-1007
- Kuhn, C.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *The ALICE experiment at LHC.* Nuclear Physics A 2007 ;Volum 787. s. 19-28
- Lunardon, M.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Open heavy flavour detection in ALICE. Nuclear physics B,* Proceedings supplements 2007 ;Volum 167. s. 25-28
- Mastroserio, A.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Examples of soft physics observables in the ALICE experiment.* Acta Physica Polonica B 2007 ;Volum 38. s. 1009-1016
- Monteno, M.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *The physics programme of the ALICE experiment at the LHC.* Nuclear Physics A 2007 ;Volum 782. s. 283-290
- Morsch, A.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Hard probes capabilities of ALICE: Jets and photons.* Nuclear Physics A 2007 ;Volum 783. s. 427-434
- Nappi, E.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Progress of the ALICE experiment.* Acta Physica Polonica B 2007 ;Volum 38. s. 313-326
- Noriega, M.L.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Jet physics in ALICE.* European Physical Journal C 2007 ;Volum 49. s. 315-319
- Oyama, K.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Tveter, Trine Spedstad; ALICE, Collaboration; Skaali, Toralf Bernhard; Vik, Thomas; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Physics with ALICE transition radiation detector.* Acta Physica Polonica B 2007 ;Volum 38. s. 1017-1024
- Richter, M; Aamodt, Kenneth; Alt, T; Bablok, S; Cheshkov, C; Hille, Per Thomas; Lindenstruth, V; Ovrebekk, G; Ploskon, M; Popescu, S; Rohrich, D; Steinbeck, TM; Thader, J. *High level trigger applications for the ALICE experiment.* IEEE Transactions on Nuclear Science 2008 ;Volum 55.
- Samset, Bjørn Hallvard; Pajchel, Katarina; Hille, Per Thomas. *Jakten p ursmellet.* Aftenposten [Newspaper] 2007-06-20
- Snellings, R.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Anisotropic flow from RHIC to the LHC.* European Physical Journal C 2007 ;Volum 49. s. 87-90

- Sommer, W.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration. *Quarkonia Measurements with the Central Detectors of ALICE.* Quark Matter 2006; 2006-11-14 - 2006-11-20
- Turrisi, R.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration; Cortese, P.; Dellacasa, G.; Ramello, L.; Sitta, M.; Ahmad, N.; Ahmad, S; Ahmad, T.; Bari, W.. *Perspectives for the measurement of beauty production via semileptonic decays in ALICE.* Acta Physica Polonica B 2007 ;Volum 38. s. 1039-1046
- Zampolli, C.; Bravina, Larissa; Hille, Per Thomas; Løvhøiden, Gunnar; Skaali, Toralf Bernhard; Tveter, Trine Spedstad; Vik, Thomas; ALICE, Collaboration. *Heavy ion physics with the ALICE experiment at LHC.* 42nd Rencontres de Moriond on QCD and Hadronic Interactions; 2007-04-16 - 2007-04-24
- Bablok, Sebastian Robert et. al *High level trigger online calibration framework i ALICE*, J. Phys. Conf. Ser. 2008; volume 119,
- ∼120 publications as member of the ALICE collaboration.

Appendix B

Some Properties of the $PbWO4$ Crystal

Radiation length The radiation length of the $PbWO4$ crystals can be estimated using the the formula [46]

$$L_{rad} = \frac{716.4 g/cm^2 A}{Z(Z+1)\ln(287/\sqrt{Z})} \tag{B.1}$$

which gives an estimate with an accuracy within 2.5% for all elements except helium [46]. L_{rad} is in units of mass thickness, A is the atomic weight and Z is the atomic number. For compounds the radiation length can be estimated using Bragg's rule

$$\frac{1}{L_{rad}} = w_1(\frac{1}{L_{rad}})_1 + w_2(\frac{1}{L_{rad}})_2 + \cdots \tag{B.2}$$

where $w_1 w_2 \cdots$ are the fraction by weight of each element in the compound given by

$$w_i = \frac{a_i A_i}{Am} \tag{B.3}$$

Here a_i is the number of atoms of the ith elements in the molecule, A_i the atomic mass of compound i and A_m total atomic mass of the molecule given by

$$A_m = \sum a_i A_i \tag{B.4}$$

For $PbWO4$ crystals this yield

$$A_m^{PbWO_4} = A_{Pb} + A_W + 4A_O = 207.2 + 183.84 + 15.99 = 455.03 \tag{B.5}$$

$$w_{Pb} = \frac{A_{Pb}}{A_m^{PbWO_4}} = \frac{207.2}{455.03} = 0.45535 \tag{B.6}$$

$$w_W = \frac{A_W}{A_m^{PbWO_4}} = \frac{183.84}{455.03} = 0.40402 \tag{B.7}$$

$$w_O = \frac{4A_O}{A_m^{PbWO_4}} = \frac{15.99}{455.03} = 0.14064 \tag{B.8}$$

The radiation length of the individual compounds, using B.1, yields

$$L_r^{Pb} = \frac{716.4 g/cm^2 A_{Pb}}{Z(Z+1)ln(287/\sqrt{Z})} = \frac{716.4 g/cm^2 207.2}{82(82+1)ln(287/\sqrt{82})} = 6.3105 g/cm^2 \quad \text{(B.9)}$$

$$L_r^{W} = \frac{716.4 g/cm^2 A_{W}}{Z(Z+1)ln(287/\sqrt{Z})} = \frac{716.4 g/cm^2 183.84}{74(74+1)ln(287/\sqrt{74})} = 6.7656 g/cm^2 \quad \text{(B.10)}$$

$$L_r^{O} = \frac{716.4 g/cm^2 A_{O}}{Z(Z+1)ln(287/\sqrt{Z})} = \frac{716.4 g/cm^2 15.999}{8(9)ln(287/\sqrt{8})} = 34.4600 g/cm^2 \quad \text{(B.11)}$$

Using Equation B.2 for the compound one obtains

$$L_{rad}^{PbWO_4} = 7.36 g/cm^2 \quad \text{(B.12)}$$

Or in centimeter by dividing by the mass density

$$L_{rad_{cm}}^{PbWO_4} = \frac{7.36 g/cm^2}{8.28 g/cm^3} = 0.89 \quad cm \quad \text{(B.13)}$$

Moliere radius The *Moliere* radius is defined as the cylinder containing 90 % of the total shower energy for an electromagnetic shower. The $PbWO_4$ crystal has relatively small Moliere radius of ∼2 cm. The small Moliere radius was one of the most important reasons why the $PbWO_4$ crystal was chosen. The transverse size of 2.2x2.2cm^2 of the crystals are chosen in order to match the Moliere radius.

Table B.1: Some properties of the PWO crystal [14].

Density	8.28 g/cm^3
Radiation length	0.89 cm
Moliere radius	2.0 cm
Melting point	1123 $°C$

Summary

Appendix C

Profiling and Optimization of PHOS HLT Software Performance using VTUNE

"We should forget about small efficiencies, say about 97 % of the time: premature optimization is the root of all evil. (Donald E. Knuth)". A software solution that is optimal from the point of view of performance often requires the software to be written in a way that makes the source code difficult to understand, thereby adding to the cost of software maintenance. In order to facilitate software maintenance optimization should be done only on the part of the source code that can be expected to significantly improve the overall performance. One should wait until it is really necessary to improve the performance in order not to slow down the development process in the early stage.

Several profiling tools was tested and considered and it was found that *VTUNE* from Intel was the profiler best suited to the task. VTUNE is commercial software but can be downloaded for free from the Intel homepage provided that it is stated that the profiler will be used only for development of non commercial software.

One issue of using VTUNE is that it can be used only on PCs running Intel CPUs whereas the HLT cluster runs on AMD CPUs. The performance can of course be different on Intel based versus an AMD based PCs and a piece of software optimized for and Intel CPU might not be perfectly optimized for an AMD. However it is often the case that bottlenecks in the performance is caused by obvious mistakes that would kill the performance on any platform. The general procedure adopted for PHOS has been to optimize the the code on an Intel PC and then run the benchmarking of the optimized software on the HLT cluster at ALICE.

The running of the HLT code for any detector including PHOS can be divided in three steps. The initialization, the event loop and de-initialization phase. These steps are represented in the software components by the three function calls *DoInit(...)* and *DoDeInit(...)* which are called once at start of run and end of run respectively and

Function	CPU%	Caller
DoEvent	67.07	The HLT framework
Evaluate	31.00	DoEvent
GetEnergy	0.21 %	Evaluate
GetTiming	0.21 %	Evaluate

Table C.1: Function view of the VTUNE output.

a *DoEvent(....)* function which is called for every event. Typically most of the total processing time is used in the event loop. This simplifies optimization since the *DoEvent* function always calls the same processing code. As an example the component that handles the raw data processing for PHOS calls 1) the decoder (see Appendix D) to decode the data, 2) the mapping to translate from hardware to geometrical addresses, 3) the algorithm to evaluate the energy and TOF from the raw data. VTUNE has several features for debugging and profiling and a description of these features is outside the scope of this text. The feature most extensively used for PHOS has been the CPU sampling. VTUNE reads from the CPU stack which function is executing at any given time. If debugging symbols are present in the compiled code VTUNE will be able to determine the name of the function, which library it is loaded from and what process was running the code at the time the sample was taken. One of the advantages with VTUNE is that this sampling is done completely transparently to the user application being profiled with a relatively small overhead in terms of additional processing time. Any number of processing threads might be running simultaneously. In addition the profiling is independent of the programming language.

VTUNE samples the stack of the CPU to determine which function was executing when the samples were taken. The sampling is done typically several million times to obtain statistics of the most frequently called functions.

The optimization is done by simply starting on the top of the list of functions spending the most of the CPU time and optimize function by function until the overall performance is satisfactory.

C.1 Benchmarking Results

It was found that around 67 % of the CPU time was spent on decoding the data and 26 % on analyzing it. It was concluded that the decoder available in AliRoot at that time was too slow and that a new faster decoder was needed in order for the overall performance to satisfy the requirements. Some of the results from the profiling are shown in Table C.1 and C.2. The first *Next* function in the table is called for each event. The main performance problem that the offline decoder was implemented as streamer where the entities of the stream is individual ALTRO samples represented by the second *Next* function in the table

Function name	CPU%	Caller
Next	30.12	The off-line decoder
GetNextWord	23.93	Next (the second one)
Next	21.93 %	Next (the first one)
ReadTrailer	0.93 %	Next (the first one)

Table C.2: Fractional CPU times spent by functions called by the DoEvent function. The list is not complete and the total amount is therefore less than 100%.

which is called for each sample. This function calls directly or indirectly more than 20 other functions and performs a lot of tests. The solution to the performance problem was to remove this sample streamer altogether and replace it by a stream of ALTRO payloads. The resulting new fast decoder is described in Appendix D. After replacing the standard off-line decoder with the new faster version the overall performance improved by a factor 10 - 15. So far the PHOS HLT chain performing the full event reconstruction has been successfully tested up to 2 KHz at the ALICE HLT cluster using simulated min bias p+p data. The processing has been tested successfully up to 7 KHz when only energy and TOF is evaluated from the raw data.

Appendix D

A Fast Decoder for PHOS, EMCAL and TPC

D.1 RCU Data Format over the DDL

The RCU data formatter converts the blocs of 4x40 bit ALTRO words into blocks of five 32 bit words. One such block will from now on be called a DDL block. In general the number of 40 bit ALTRO words of a RCU payload is not divisible by 4, therefor the last DDL data block might have either 2,3 or 4 32 bit words. It is necessary to find the position of the last ALTRO word since the RCU payload is a back linked list of ALTRO data. The position of the last 40 bit ALTRO word can be found by taking the size of the payload modulus 5. more specifically $(sizeof(ddl_buffer)/4 - headersize - Ntrailerwords) mod 5$. The mod 5 value of the DDL buffer determines unambiguous the position of the last ALTRO word and thereby the total number of ALTRO word. In addition the RCU trailer contains the count of 40 bit ALTRO words that can be used as a consistency check. Given the payload size mod 5 value the position of the last ALTRO word can be determined as given below.

- Segmentation 0: The number of 40 bit ALTRO words is divisible by 4, the last DDL block is filled according to Figure D.2(a)
- Segmentation 1; The number of 40 bit ALTRO words modulus 4 equals one, the DDL block is filled according to Figure D.2(b)
- Segmentation 2; The number of 40 bit ALTRO words modulus 4 equals two, the DDL block is filled according to Figure D.2(c)
- Segmentation 3; The number of 40 bit ALTRO words modulus 4 equals three, the DDL block is filled according to Figure D.2(d)

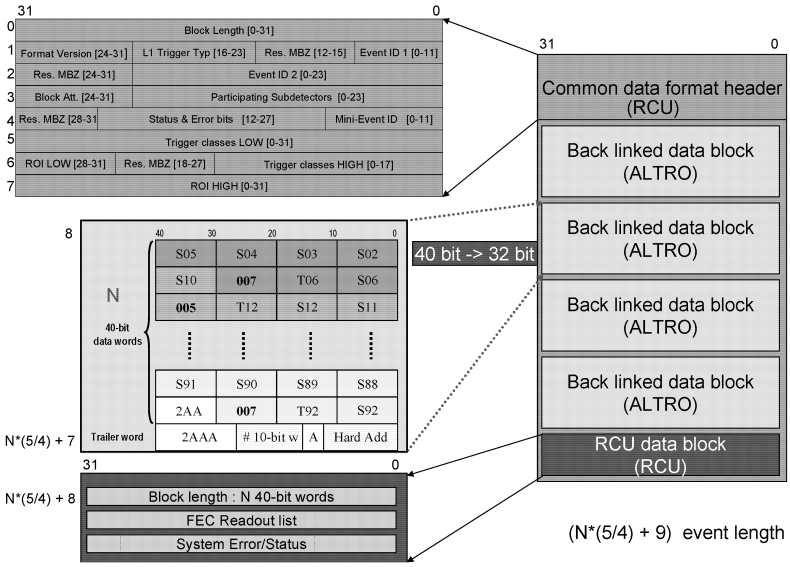
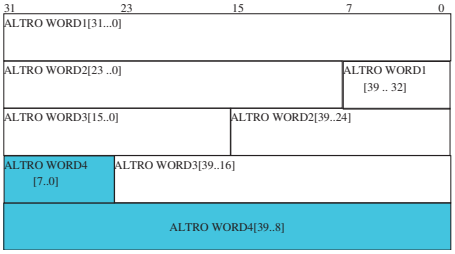


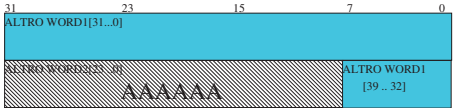
Figure D.1: ALTRO format, Figure made by Torsten Alt.

D.2 Example: Decoding of a PHOS DDL File

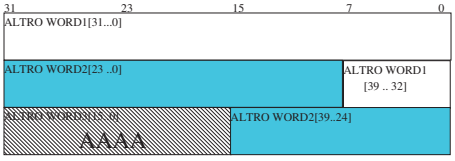
```
 [aliphoshlt@mixing cosmics_ddl]/  hexdump test.ddl

000000 ffff ffff 0000 0100 0000 0000 0000 0000
000001 0000 0000 0000 0000 0000 0000 0000 0000
000002 d435 4350 350d 50d4 0d43 d434 4350 350d
000003 50d4 0d43 d435 4350 350d 70d8 0e03 f43b
000004 c400 4610 b120 1344 444f 4531 5615 715c
000005 1605 6058 0581 5816 715c 1585 5455 c541
........

002778 82e0 2e0b e0b8 0b82 b82e 82e0 2e0b e0b8
002779 0b82 b82e c2e0 2e0b e0b8 0bc2 b82e c2e0
00277a 2e0b e0b8 0bc2 b82e c2e0 2e0b e0b8 0b82
00277b b82e 0450 4f12 48a7 aaa8 ac2b 7e52 0000
00277c
```



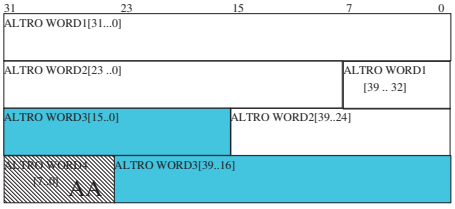
(a) Segmentation 0: Position of last ALTRO word when the a DDL block is exactly filled i.e the number of ALTRO words is divisible by 4. This corresponds to the case when N40bitwords mod 4 =0, or alternatively N32altroPayloadsize mod 5 = 0



(b) Segmentation 1: Position of the last ALTRO word of the last DDL block in the case where the last DDL block is filled with just one 40 bit word. In this case the last DDL block consists of two 32 bit words.The least significant 24 bits of the last 32 bit word of the DDL block indicated by the gray area is not used and is padded with 0xAAAAAA. This corresponds to the case when N40bitwords mod 4 = 1, or alternatively N32altroPayloadsize mod 5 = 2



(c) Segmentation 2: Position of the last ALTRO word of the last DDL block when the last DDL block is filled with two 40 bit words. In this case the the last DDL block consists of three 32 bit words. The least significant 16 bits of the last 32 bit word of the DDL block indicated by the gray area is not used and is padded with 0xAAAA. This corresponds to the case when N40bitwords mod 4 = 2, or alternatively N32altroPayloadsize mod 5 = 3



(d) Segmentation 3: Position of the last ALTRO word of the last DDL block when the last DDL block is filled with three 40 bit words. In this case the the last DDL block consists of four 32 bit words. The least significant 8 bits of the last 32 bit word of the DDL block indicated by the gray area is not used and is padded with 0xAA. This corresponds to the case when N40bitwords mod 4 = 3, or alternatively N32altroPayloadsize mod 5 = 4

Figure D.2: Four possible ways of filling the last DDL block.

Lets try to figure out which segmentation is used for the last DDL block of this RCU payload.

```
[aliphoshlt@mixing cosmics_ddl]\ ll test.ddl
-r--r--r--  1 aliphoshlt aliphoshlt 161728 May 20 16:54 test.ddl
```

We see that the file is 161728 bytes. That is 40432 32 bit words. There are one trailer word and 8 common data header words. The payload then consists of 40432 -(8+1) = 40423 32 bit words. 40423 mod 5 = 3 and 40423 = 5*8084 +3. In each 5x32 bit block there are 4x40 bit ALTRO words. Since 40423 mod 5 = 3 the last DDL block is filled with 2 ALTRO words. The total number of ALTRO words is then NaltroWords = 4*8084 + 2 = 32338(dec) = 0x7e52. The total number of 40 bit ALTRO words is also given by the RCU trailer. As a consistency check one can verify that the number of 40 bit words given by the RCU trailer is indeed 0x7e52. So now we try if our theories are right by

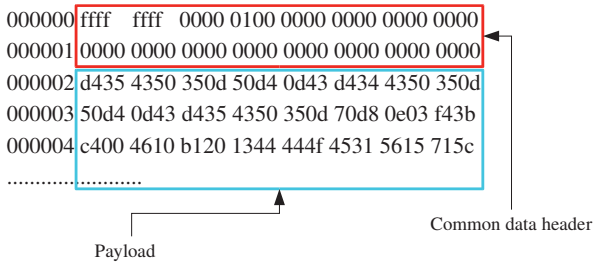


Figure D.3: Decoding the DDL files.

applying them om some real test data taken with the PHOS detector (cosmic). Looking again at the raw data from the previous section we fill in the 32 bit words and do some bit banging to see if it everything makes sense.

D.3 Implementation

For raw data coming from the RCU it is natural to treat the payload as consisting of 160 bit *ddl words* since the the data formatter of the RCU packs four 40 bit words into five 32 bit words so that the decoding of any 160 bit is always the same. Secondly the natural entities of the payload is, referring to Figure D.6 ALTRO payloads and bunches and not individual samples. The decoding is performed in two steps. The number of 160 bit words contained in the payload coming from one event and from one RCU is determined from the size of the data. The procedure outlined in Section D.2 is implemented with the

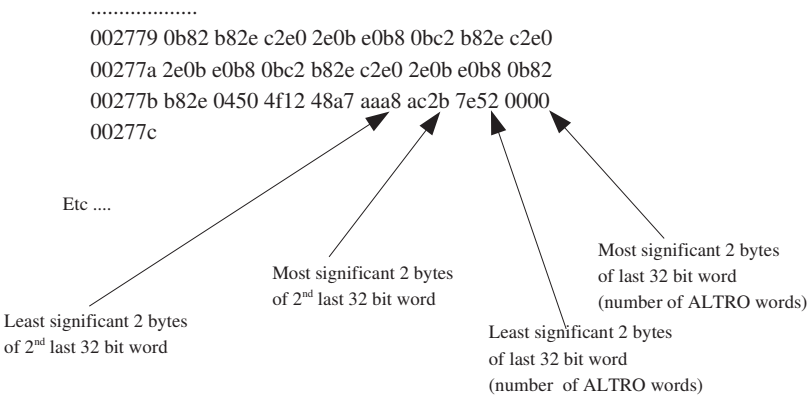


Figure D.4: Decoding the DDL files.

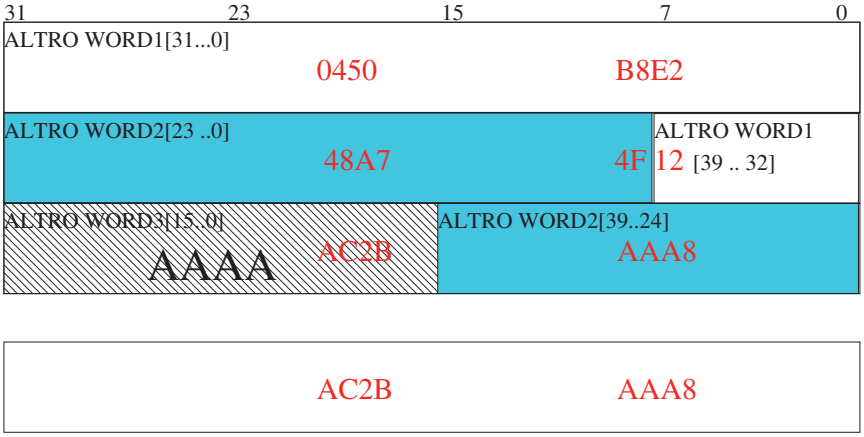


Figure D.5: Decoding the DDL files.

following 41 lines of code shown below which is called once for each DDL word. If the payload does not align to 160 bits the last word is padded with zeroes. In the second step

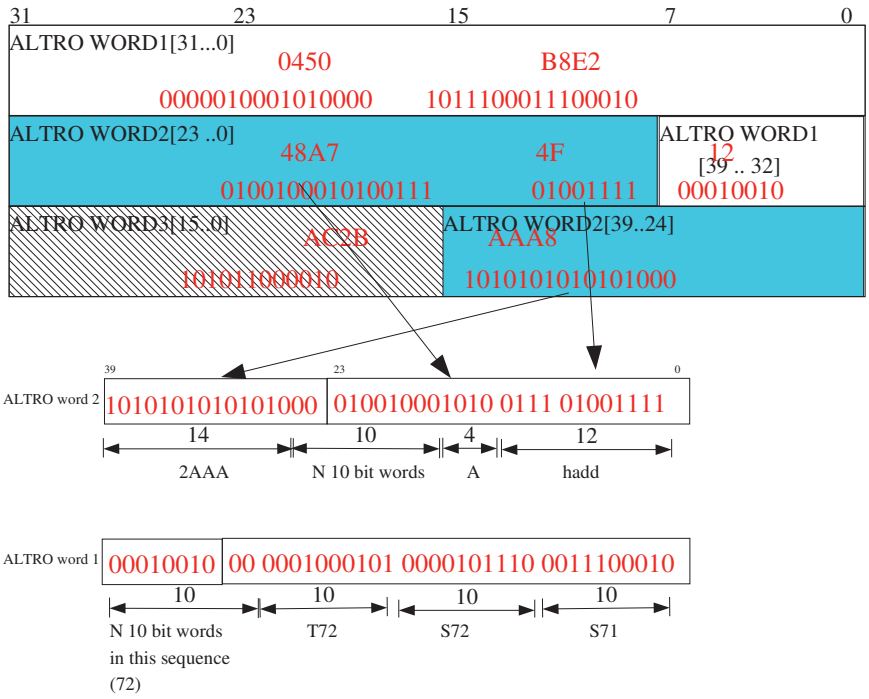


Figure D.6: Decoding the DDL files.

the ALTRO payloads and bunches provided to the user application looping through the back linked list starting from the end of the decoded payload as shown in Figure D.6. An abstraction of the interface is shown below

```
 while( fDecoderPtr->NextChannel(fAltroDataPtr) == true )
{
          while(fAltroDataPtr->NextBunch(fAltroBunch)
              {
                 //Do something with the data
              }
       }
```

D.4 Benchmarking Results

The benchmarking was done on several different types of data. Real, simulated, zero suppressed, non zero suppressed, PHOS, TPC data. It was found to be 10 - 15 times faster than the previous decoder used by ALICE offline. The new decoder satisfies all the requirements for the HLT.

D.5 The Decoding Algorithm

```
void AliAltroDecoder::DecodeDDLBlock()
{
  fOutBuffer[fOutBufferIndex] =  *f32DtaPtr & 0x3ff;  //s0
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] = (*f32DtaPtr & 0xffc00) >> 10; //s1
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] = (*f32DtaPtr & 0x3ff00000) >> 20; //s2
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] = (*f32DtaPtr & 0xc0000000) >> 30; //s3_1
  f32DtaPtr ++;
  fOutBuffer[fOutBufferIndex] =  fOutBuffer[fOutBufferIndex] | ((*f32DtaPtr & 0xff) << 2); //s3_2
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0x3ff00) >> 8; //s4
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xffc0000) >> 18; //s5
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xf0000000) >> 28; //s6_1
  f32DtaPtr ++;
  fOutBuffer[fOutBufferIndex] =  fOutBuffer[fOutBufferIndex] | ((*f32DtaPtr & 0x3f) << 4); //s6_2
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xffc0) >> 6; //s7
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0x3ff0000) >> 16; //s8
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xFC000000) >> 26; //s9_1
  f32DtaPtr ++;
  fOutBuffer[fOutBufferIndex] =  fOutBuffer[fOutBufferIndex] | ((*f32DtaPtr & 0xf) << 6); //s9_2
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0x3ff0) >> 4; //s10
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xffc000) >> 14; //s11
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xff000000) >> 24; //s12_1
  f32DtaPtr ++;
  fOutBuffer[fOutBufferIndex] =  fOutBuffer[fOutBufferIndex] | ((*f32DtaPtr & 0x3) << 8); //s12_2
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xffc) >> 2; //s13
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0x3ff000) >> 12; //s14
  fOutBufferIndex ++;
  fOutBuffer[fOutBufferIndex] =  (*f32DtaPtr & 0xffc00000) >> 22; //s15
  f32DtaPtr ++;
  fOutBufferIndex ++;
}
```

Appendix E

Methods of Analysis for Noise Measurements

This chapter describes methods of noise analysis for *pedestal runs*. *pedestal run* means that data is taken without physics trigger and without zero suppression.

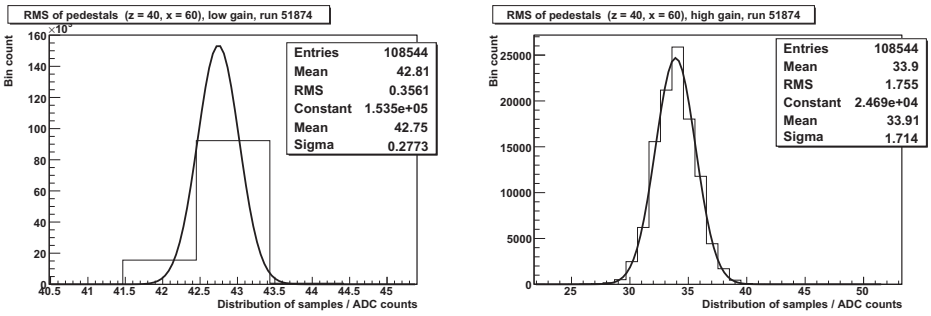
E.1 Noise measurements

The noise levels for *healthy* channels is typically significantly less than the ADC quantization error for low gain channels and in the order of the ADC quantization error for high gain channels. The therm *healthy* in this context is used for channels where the noise is not unexpectedly high and is dominated by electronics and APD noise. In general the noise is dominated by electronic noise from the shapers for the low gain channels and by the APD and CSP noise for high gain channels. In either case the noise is assumed to be Gaussian however after ADC quantization, the distribution of the digitized samples follows a Bernoulli distribution rather than a Gaussian distribution. In addition, during *noisy* conditions as described in Chapter 9 excessive/unexpectedly high noise is typically associated with with one or a few single frequency components that sits on top of the Gaussian noise. For a Bernoulli random distribution the RMS is *not* a consistent estimate of the standard deviation of the underlying signal (i.e the actual signal before quantization). For example, a baseline signal with a standard deviation much smaller than the ADC quantization error that sits exactly between two ADC quantization levels will have a RMS of 1/2 after quantization regardless of the standard deviation whereas the same signal sitting in the middle of 2 ADC quantization levels will have an RMS of zero. However, assuming that the underlying signal is Gaussian, the probability that a given sample ends up in a given ADC quantization bin can be described as a Bernoulli random process where the outcome of the individual trials is given by standard deviation and the mean value of the underlying signals. A consistent estimate of the standard deviation of

the noise can therefore be obtained by making a histogram of all the samples, typically accumulated over several events, with one bin for each ADC quantization level and fitting the resulting histogram with a Gaussian.

E.1.1 Rejecting Fit Results Based on Fit Probability

The fit probability assuming a Gaussian distribution is evaluated and if the fit probability is less than < 0.3 the standard deviation obtained from the fit is rejected and the RMS is used instead as an estimate of the noise. Figure E.1(a) shows a typical histogram for a



(a) Typical pedestal histogram low gain

(b) Typical pedestal histogram high gain

Figure E.1: The figure shows the distribution of samples accumulated over 212 events, each event having 512 samples for each of low and high gain respectively.

typical low gain channel. As indicated in the figure the RMS is 0.36 ADC counts whereas the standard deviation obtained from the Gaussian fit is 0.28, illustrating that the RMS is not a consistent estimate of the standard-deviation when the standard deviation of the noise is comparable to the ADC quantization error. Figure shows the same channel after high gain amplification. For the high gain in this case the RMS and the standard deviation obtained from the Gaussian fit is consistent.

E.2 Fourier Analysis

E.2.1 Power Spectrum Estimation

In order to facilitate the estimation of the power spectrum from the data the following assumptions are made.

- The time dependent signal for an individual channel is assumed to be a Wide Sense Stationary (WWS) process

- For different events, the time dependent signal is assumed to be the independent realizations of the same WWS process

The power spectrum is estimated via the auto-correlation defined as

$$r_x(k,l) = E\{x(k)x^*(l)\} \tag{E.1}$$

where $x(k)$ and $x(l)$ denotes the digitized signal at index k and l respectively. For a WWS process the expectation value $Ex(k)x^*(l)$ depends only on the time difference $l-k$ and we write the auto-correlation as [61]

$$r_x(k) = E\{x(n)x^*(n-k)\} \tag{E.2}$$

where $x(k)$ and $x(l)$ denotes the digitized signal at index k and l respectively. For a WWS process the expectation value $Ex(k)x^*(l)$ depends only on the time different $l-k$ and we write the auto-correlation as [61]

$$r_x(k) = E\{x(n)x^*(n-k)\} \tag{E.3}$$

The *power spectral density* or *power spectrum* is defined as the Fourier transform of the auto-correlation given by the equation [61]

$$P_x(e^{j\omega}) = \sum_{k=-\infty}^{\infty} r_x(k)e^{jk\omega} \tag{E.4}$$

Referring to equation E.4, estimating the power spectrum is equivalent to estimating the auto-correlation. $r_x(k)$ is typically not known and has to be estimated from the data. This is also the case for PHOS detector data. The auto-correlation can be estimated from the data using the equation

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1-k} x(n+k)x^*(n); k = 0, 1, ..., N-1 \tag{E.5}$$

where N is the number of samples. In Equation E.5 it is implicitly assumed that the signal is zero outside the sampling window. Equation therefore gives a biased estimate of $r_x(k)$ given by.

$$E\hat{r}_x(k) = \frac{N-k}{N} r_x(k) \tag{E.6}$$

i.e the auto-correlation multiplied with Bartlet triangular window w_B defined as.

$$W_B = \left\{ \begin{array}{rcl} \frac{N-|k|}{N} & ; & |k| \leq N \\ 0 & ; & |k| > 0. \end{array} \right\}$$

The estimate of the power spectrum using Eq E.5 is commonly denoted the *periodogram*

$$\hat{P}_{per} = \sum_{k=-N+1}^{N-1} \hat{r}(k)e^{-j\omega} \tag{E.7}$$

Using the convolution theorem the expected value of the periodogram in the frequency domain is the power spectrum convoluted with the Fourier-Transform of a Bartlett window

$$E\hat{P}_{per}(e^{j\omega}) = \frac{1}{2\pi} P_x(e^{j\omega}) * W_B(e^{j\omega}) \tag{E.8}$$

where

$$W_B(e^{j\omega}) = \frac{1}{N}\left[\frac{sin(N\omega/2)}{sin(\omega/2)}\right]^2 \tag{E.9}$$

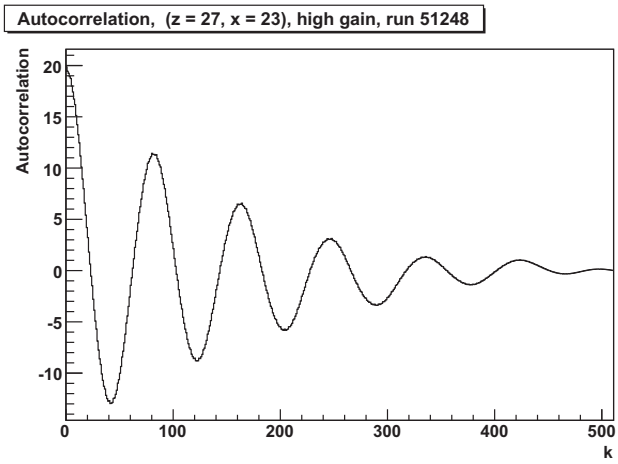
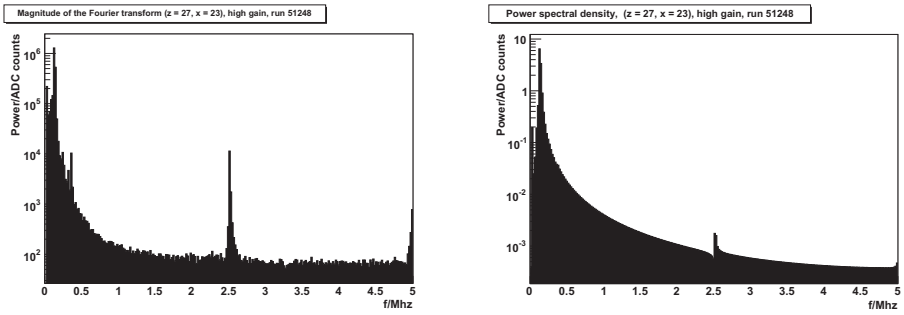


Figure E.2: Typical auto-correlation for for a channel contaminated by sinus noise.

E.2.2 Estimating the Power Spectrum from Real PHOS Detector Data

To the first approximation the power-spectrum is proportional to the magnitude of the Fourier transform. A comparison was made between the average magnitude of the Fourier transform and the power spectrum. Figure E.3 shows a comparison of the power spectrum estimated as the average magnitude of the Fourier transform and the average *peridogram*



(a) Average magnitude of the Fourier transform

(b) Power spectrum

Figure E.3: Comparison between the power spectrum and the average magnitude of the Fourier transform.

as defined by Equation E.7. As indicated in the figure the periodogram seem to give a more precise estimate of the power spectrum. Referring to Eq.E.6 the bias of the periodogram is small if N is large and if rk falls of rapidly to zero. In addition the Bartlett window approaches and impulse in the frequency domain for large N. Given the above the approach for estimating the power spectrum is the following:

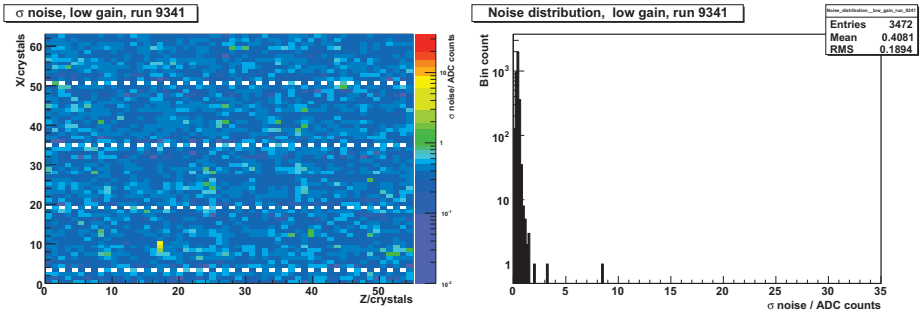
- The electronics is configure to sample as many time bins as possible for each pedestal trigger in order to increase the frequency resolution and to minimize the bias of the *periodogram*
- The auto-correlation is estimated as accurately as possible i.e averaged over many events
- The *periodogram* is evaluated for the average correlation obtained above

For most of the pedestal runs the electronics was configured to sample the maximum[1] 512 samples. Figure E.2 of a typical auto-correlation sequence shows that the correlation approaches zero for k $\sim$500.

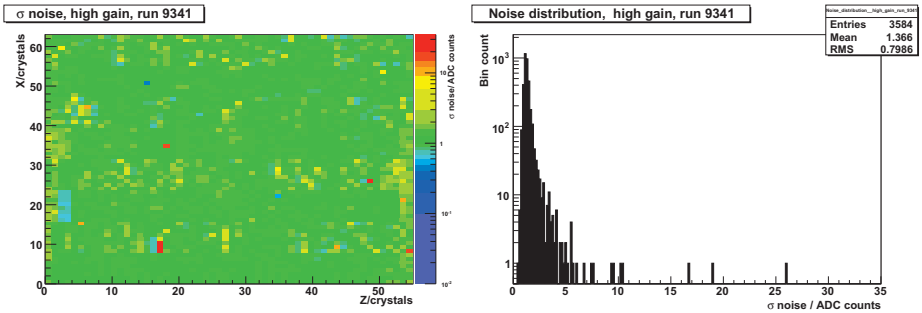
[1]The maximum number of samples is actually 1023, but the ALTRO chips used for PHOS was only tested for a maximum of 512 samples

Appendix F

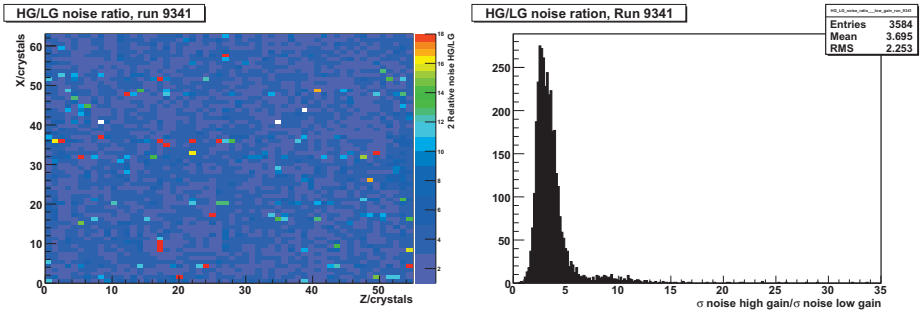
Additional Noise Measurement Results for Section 9.2.3



(a) Low gain

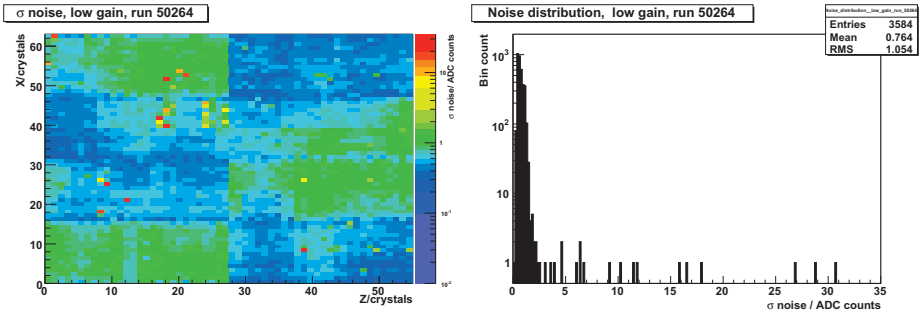


(b) High gain

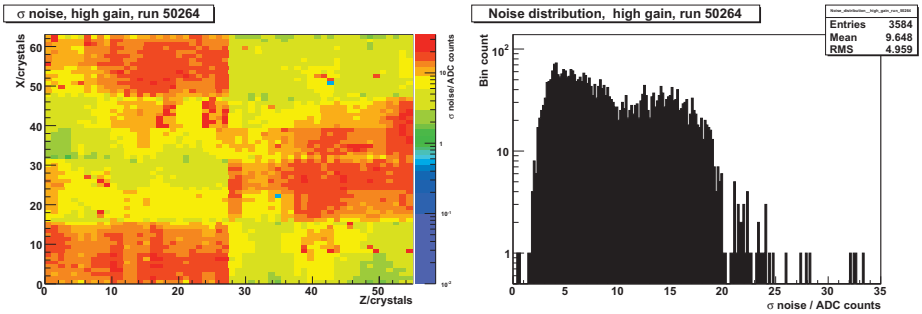


(c) High gain/low gain noise ratio

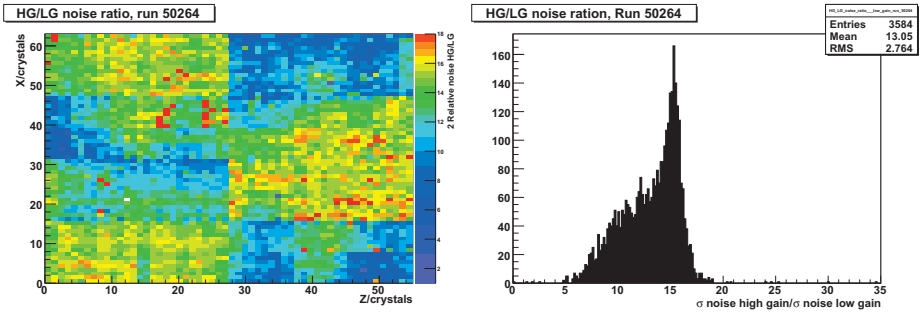
Figure F.1: Run 9341: HV on, Gain 20 (Data taken in the PHOS lab 21/4/2008)



(a) Low gain

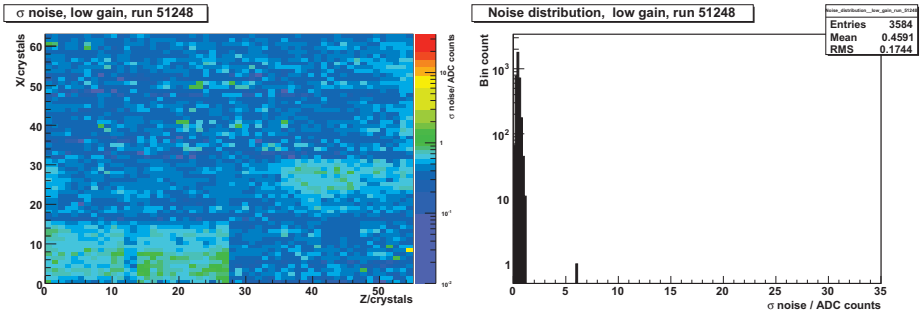


(b) High gain

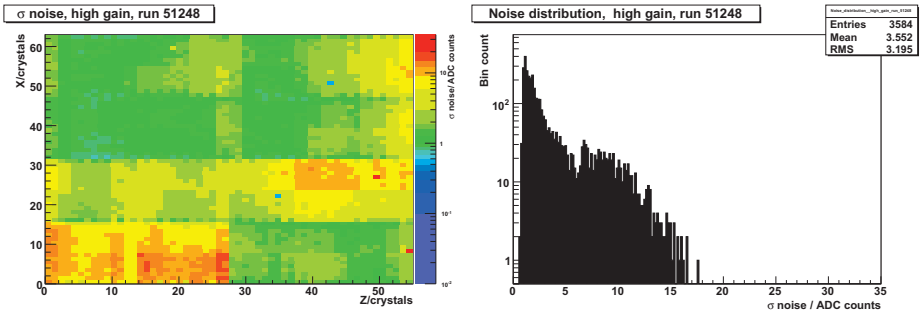


(c) High gain/low gain noise ratio

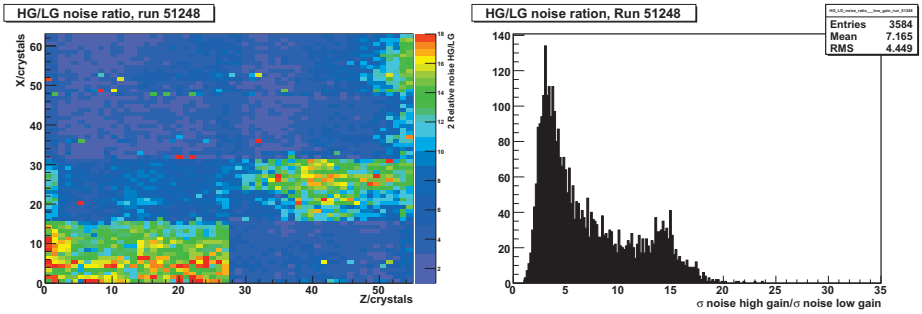
Figure F.2: Run 50264: Gain 50 (Data taken at P2 5/8/2008).



(a) Low gain

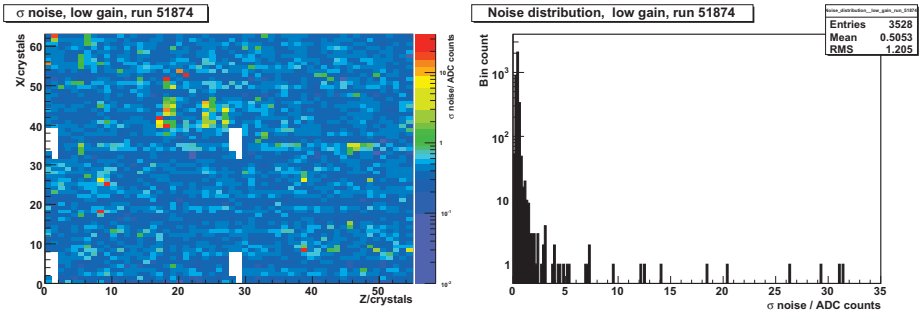


(b) High gain

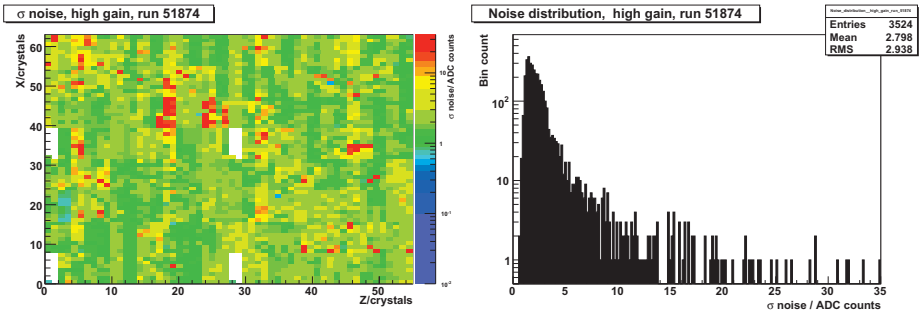


(c) High gain/low gain noise ratio

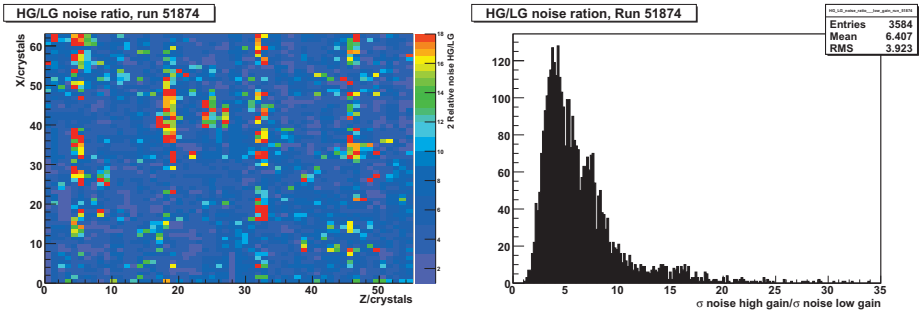
Figure F.3: Run 51248: HV on, no bias (Data taken at P2 7/8/2008)



(a) Low gain

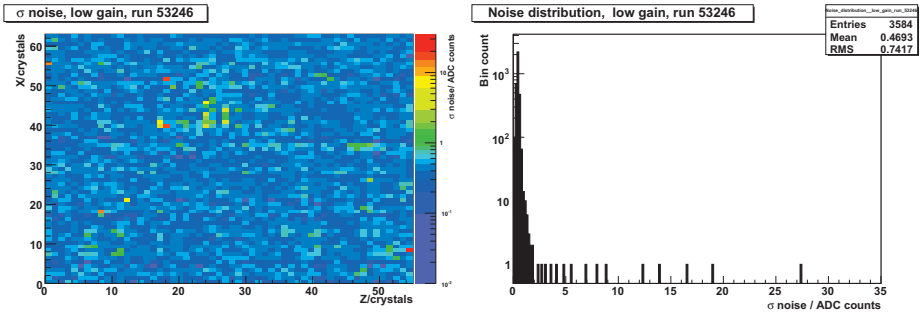


(b) High gain

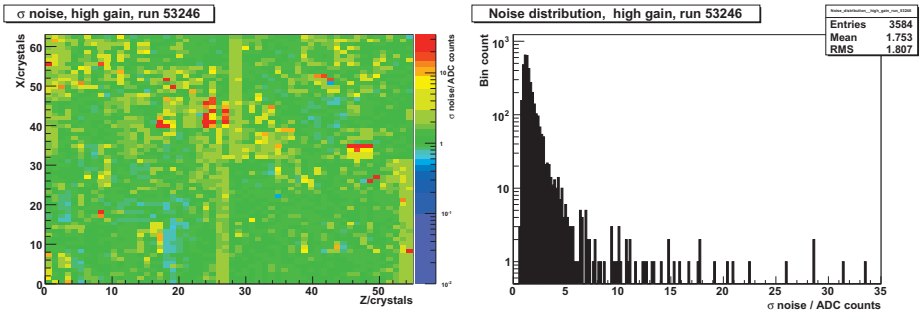


(c) High gain/low gain noise ratio

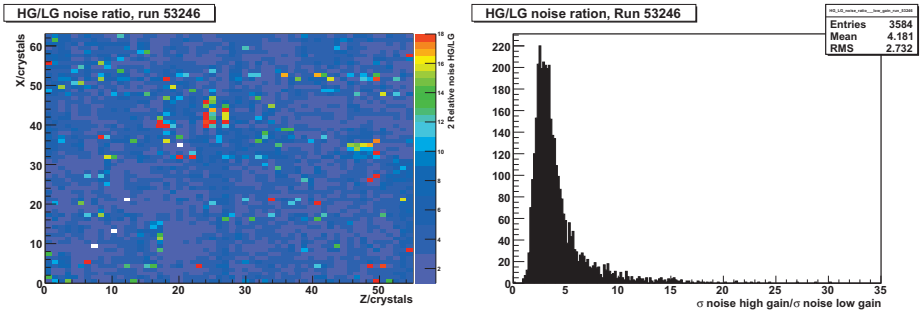
Figure F.4: Run 51874: Gain 50 at room temperature (data taken at P2 13/8/2008)



(a) Low gain



(b) High gain



(c) High gain/low gain noise ratio

Figure F.5: Run 53246: Gain 30 at room temperature (data taken at P2 21/8/2008)

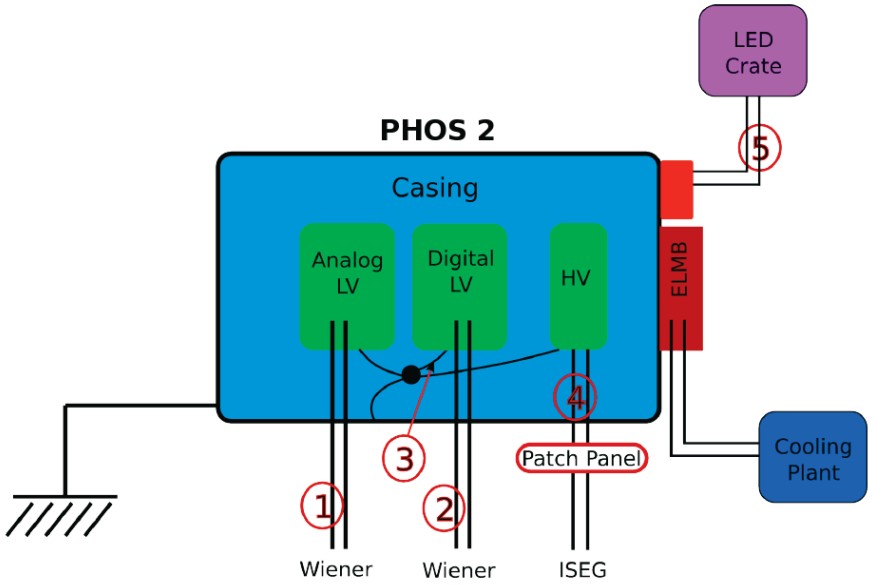


Figure F.6: Analogue noise measurements at P2 for the second PHOS module. The noise was measured by placing a coil around five different measurement points: 1) Analog ground of the Wiener PS, 2) Digital ground of the Wiener PS, 3) Noise of the digital ground cable to the PHOS casing, 4) ISEG HV cable, 5) Flat cable connecting the LED crate and the LEDs.

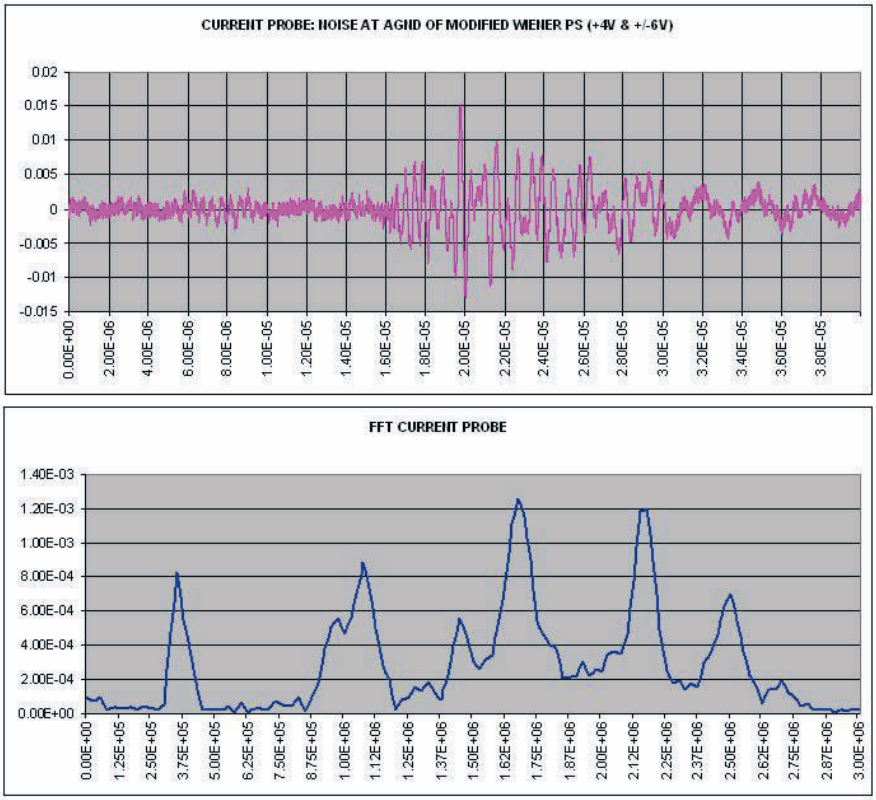


Figure F.7: Noise measured at the analogue ground (1) of the modified Wiener PS.

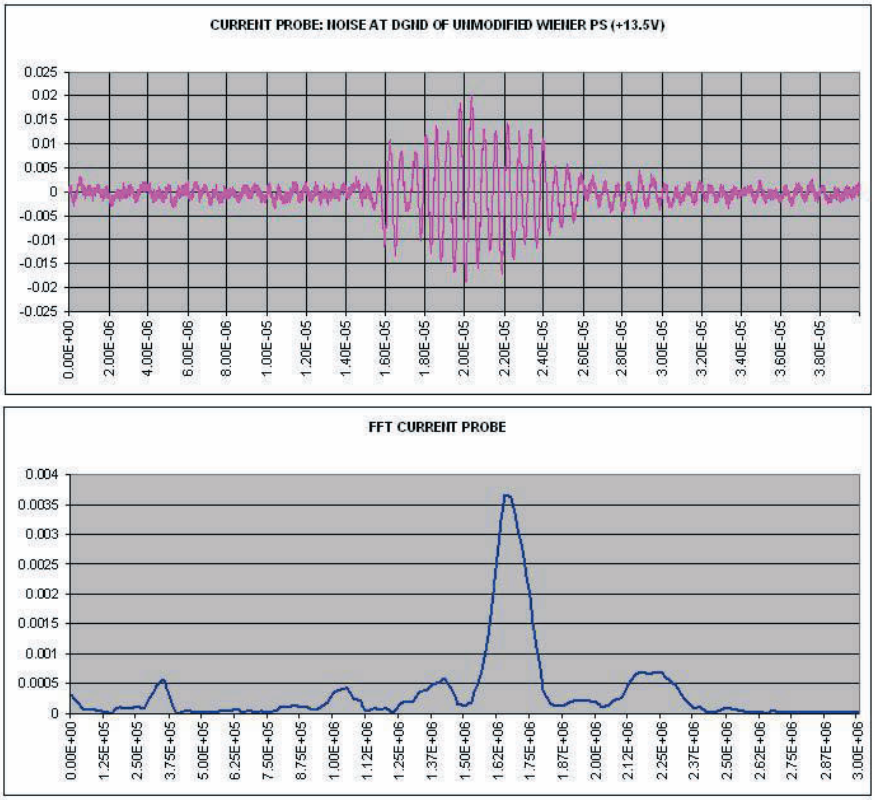


Figure F.8: Noise measured at the digital ground (2) of the non-modified Wiener PS.

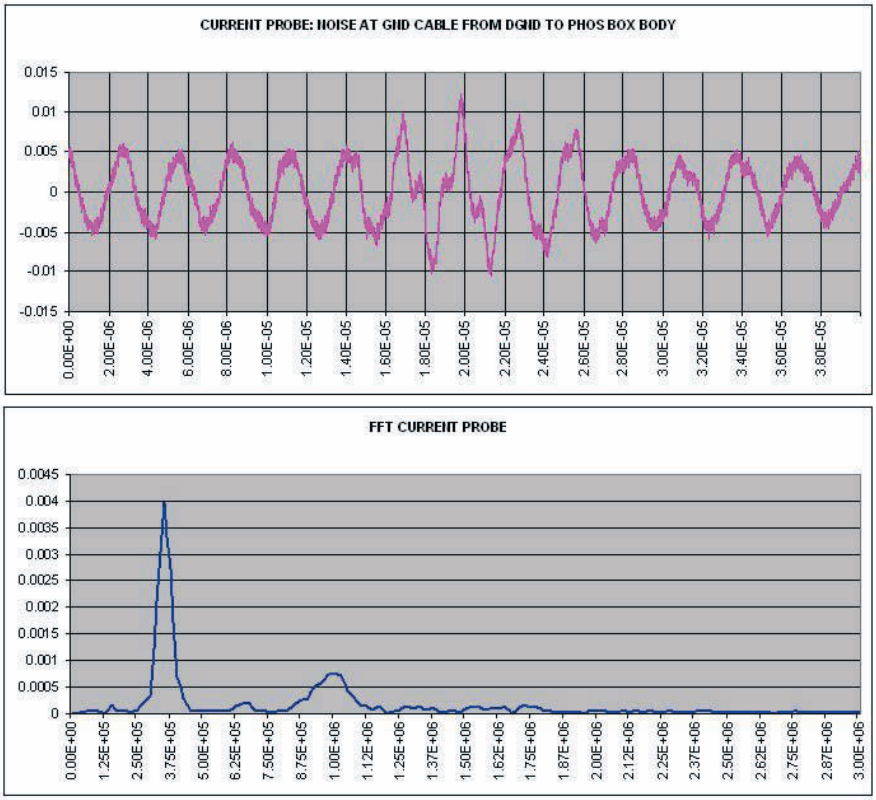


Figure F.9: Noise measured at the digital ground (3) to the PHOS casing.

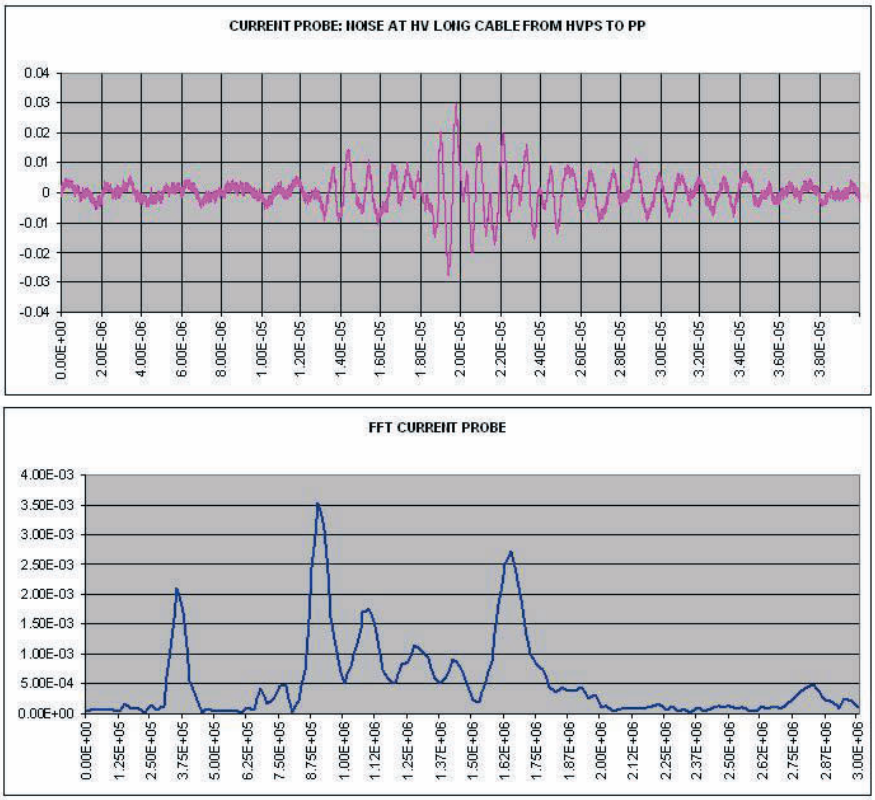


Figure F.10: Noise measured at the HV short cables between the ISEG PS and the patch panel (4), (See Figure F.6).

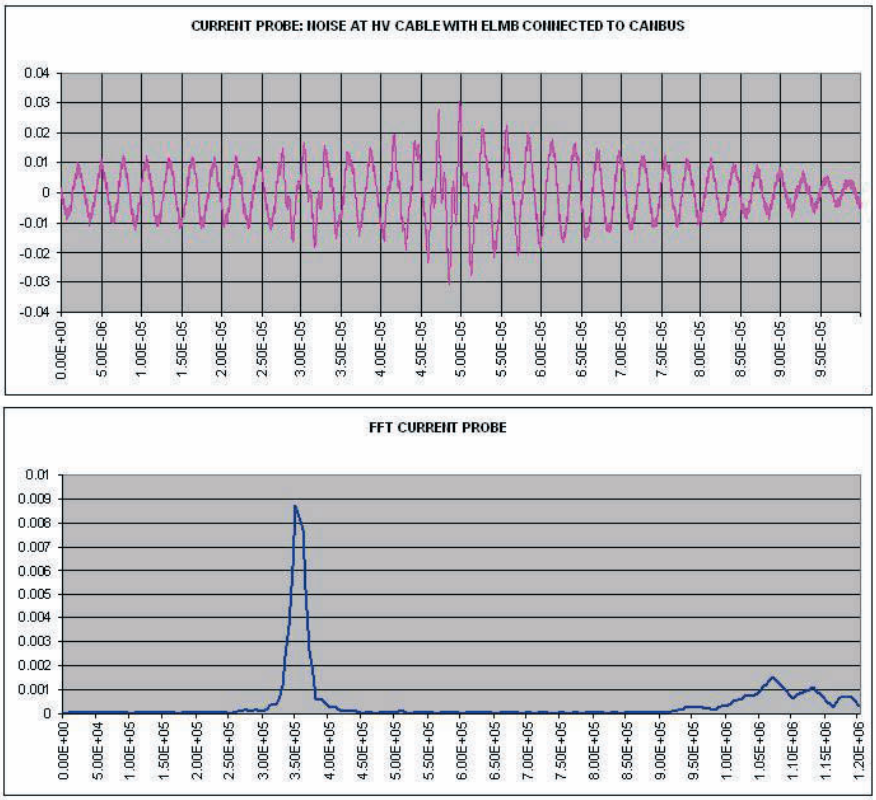


Figure F.11: Noise measured at the HV long cables between the patch panel and PHOS (4) with ELMB connected to CANBUS (See Figure F.6).

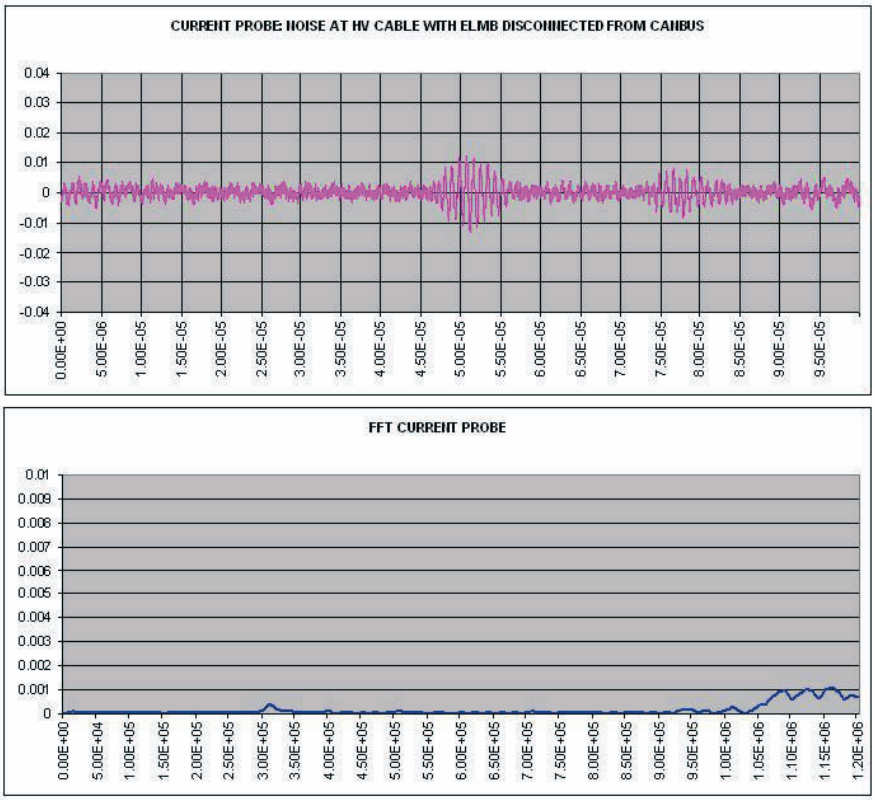


Figure F.12: Noise measured at the HV long cables between the patch panel and PHOS (4) with ELMB disconnected (See Figure F.6).

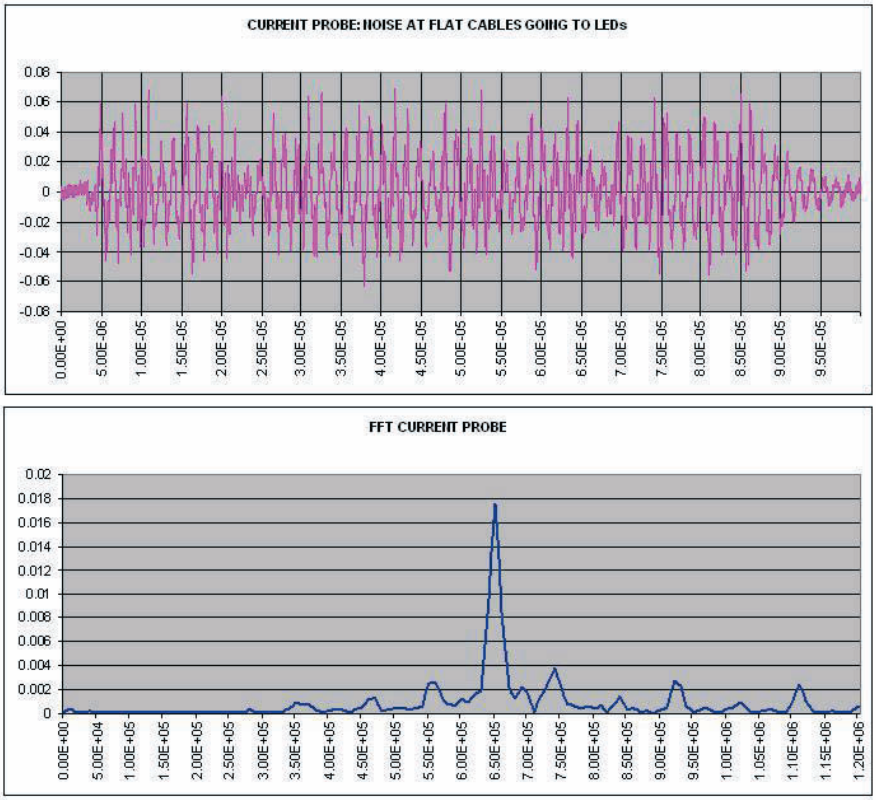


Figure F.13: Noise measured at the flat cable connecting the LED crate with the LEDs(5).

Appendix G

Properties of the Peak-Finder Algorithm and the LMS Estimator

G.1 Extraction of Amplitude and Peak Position for Individual Readout Channels

The energy and time of flight (TOF) are the only two parameters to be extracted for individual readout channels. The signal shape is assumed to be constant, i.e approximately a gamma 2, for an individual channel from event to event. It is not foreseen that any event by event pulse shape analysis will be needed.

The problem of parameter estimation can in general be described as the extraction of a set of k model parameters $\boldsymbol{\theta} = [\theta_0, \theta_1, \cdots, \theta_{k-1}]$ from a realization, or *measurement* $\boldsymbol{m} = [m_0, m_1, \cdots, m_{N-1}]$ of the N random variables $\boldsymbol{X} = [X_0, X_1, \cdots, X_{N-1}]$. We can let $\boldsymbol{X} = [X_0, X_1, \cdots, X_{N-1}]$ be spanned by the complex vector space $\mathbb{C}^N$. We could have used $\mathbb{N}^N$ since $\boldsymbol{m}$ typical are real discrete values, however we shall without loss of generality use $\mathbb{C}^N$ in order to legalize the derivation of the Peak-Finder vectors in Section G.1.5. We then have $\boldsymbol{X} \subset \mathbb{C}^N$ and $\boldsymbol{m} \in \mathbb{C}^N$. A measurement $\boldsymbol{m}$ can thereby be viewed as *point* in the *measurement space* spanned by $\mathbb{C}^N$.

Here $\boldsymbol{m}$ and $\boldsymbol{\theta}$ are related by the equation

$$\boldsymbol{m} = f(\boldsymbol{\theta}) + \boldsymbol{\epsilon}; \qquad \boldsymbol{m} \in \boldsymbol{X} \tag{G.1}$$

where f is a deterministic function that maps from the model parameters to $\boldsymbol{X}$ and $\boldsymbol{\epsilon} = [\epsilon_0, \epsilon_1, \cdots, \epsilon_{N-1}]$ is the statistical error. For a given measurement $\boldsymbol{m}$ one wants to find the set of observables $\boldsymbol{\theta}$ that maximizes the probability of observing $\boldsymbol{m}$. An estimate of $\boldsymbol{\theta}$ is obtained using an estimator $\tilde{\boldsymbol{\theta}}$, which is a deterministic function of $\boldsymbol{m}$ to obtain an *estimate* of $\boldsymbol{\theta_t}$ where $\boldsymbol{\theta_t}$ is the true value of $\boldsymbol{\theta}$.

For PHOS the free parameters are the amplitude A and the start time t_0 of the signal

so that $\boldsymbol{\theta} = [A, t_0]$ and

$$f(\boldsymbol{\theta}) = f(A, t_0) = A^2(\frac{t - t_0}{\tau_{shaper}})^2 \exp(-2\frac{t - t_0}{\tau_{shaper}}) \tag{G.2}$$

The realization $\boldsymbol{m}$ of $\boldsymbol{X}$ corresponds to N samples sampled by the ADC for one event and one readout channel.

The k parameters of $\boldsymbol{\theta}$ defines a k dimensional hyper surface, or *constraint surface* in the measurement space (see Figure G.3 for an example). The actual analysis is always done on $\boldsymbol{m}$ but the statistical properties needed to do this analysis must be inferred from $\boldsymbol{X}$. The processing of $\boldsymbol{m}$ results in an *estimator* $\tilde{\boldsymbol{\theta}}$ of the true parameters $\boldsymbol{\theta_t}$. $\boldsymbol{\theta_t}$ is also subjected to statistical fluctuations and does in general not coincide with the constraint surface. To summarize we have to deal with three different sets of model parameters when evaluating a given realization $\boldsymbol{m}$: $\boldsymbol{\theta}$ defining the constraint surface, the estimator $\tilde{\boldsymbol{\theta}}$ and the true value $\boldsymbol{\theta_t}$.

The Norm When using term *distance* in the subsequent section we should also mention with respect to which *norm*. We shall consider the $\mathscr{L}_p$ norm which is defined as $\sum_{i=0}^{n}(|x_i|^p)^{1/p}$. and the distance $d_p(x, y) = \sum_{i=0}^{n}(|x_i - y_i|^p)^{1/p}$. The most common values of p is the taxi[1] norm with p=1. For p=2 on obtains the euclidean norm and the $\mathscr{L}_\infty$ norm given by

$$\mathscr{L}_\infty = \frac{lim}{p \to \infty} \sum_{i=0}^{n}(|x_i|^p)^{1/p} = max\{|x_0|, |x_1|, \cdots, |x_n|\} \tag{G.3}$$

A minimum found with respect to one norm is in general not the same as the minimum found with respect to another. In particular, the minimization with respect to the $\mathscr{L}_2$ norm find a set of fit parameters that minimizes the squared sum of the residuals whereas the $\mathscr{L}_\infty$ finds the set of parameters that minimize the the maximum value of any given residual. For the LMS estimator the $\mathscr{L}_2$ norm is used as the name implies. For the peak-finder one wants to minimize the systematic error by minimizing the maximum distance so that in principle one should have used the $\mathscr{L}_\infty$ norm when fitting the representative plane with the representative curve (please refer to Section G.1.5). However the $\mathscr{L}_2$ is used also for the peak-finder for the following two reasons: Firstly, it simplifies the calculation of the peak-finder vectors. Secondly, the systematic errors obtained by using the $\mathscr{L}_2$ norm is already very small as presented in Appendix G and a further reduction of the systematic errors by using the $\mathscr{L}_\infty$ norm would have been insignificant in the presence of noise. In the rest of this chapter the $\mathscr{L}_2$ is therefore the only norm considered and whenever the word *distance* is used it is always with respect to the $\mathscr{L}_2$ norm.

[1]It is the distance a taxi has to drive in a rectangular grid system of streets to get from x to y

G.1.1 Signal Modeling

Any random process with Finite Impulse Response (FIR) can be modeled as the output of two (deterministic) Linear and Shift Invariant (LSI) filters denoted LSI1 and LSI2 in Figure G.1. The signal is modelled as the sum of the deterministic Dirac δ response of LSI1 and the random output from LSI2. We need a random input in order to produce a random output from a deterministic filter. The statistical errors are therefore modelled as the Gaussian white noise response of LSI2. The ADC can be modeled as a filter acting on the sum LSI1 + LSI2. This filter is non linear since the ADC introduces and round-off error which is independent of the amplitude of the signal. It should be pointed out here that the representation of the random process so presented is just a logical model. Physically, LSI, and LSI is typically the same filter, the shaper for PHOS filters of course both the noise and the signal. Furthermore the noise on the input of the shapers are not necessarily Gaussian white noise. For PHOS the δ response of LSI1 would be the time dependent signal given by Equation 4.13 and the white noise response of LSI2 the joint probability density of the noise.

The result of the ADC digitization is to produce a sequence discrete time signal from the continuous output of LSI1 + LSI2. This output corresponds to $\boldsymbol{m}$ being a realization of $\boldsymbol{m} = [m_0, m_1, \cdots, m_{N-1}]$ as described in Section G.1. Here any $m_i \in \boldsymbol{m}$ can take an value in the range $[0, \cdots, 1023]$ since the ALTRO ADC is 10 bit. Equation 4.13 is in the context of the following discussion the *signal model* in the case of PHOS, but any signal model might be substitute in the rest of this chapter.

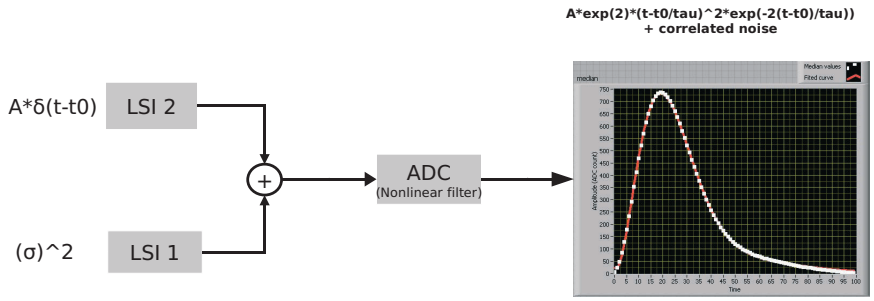


Figure G.1: Representation of a generalized signal model in therms of two Liner and Shift Invariant (LSI) filters and one nonlinear filter representing the ADC.

G.1.2 The Choice of Free Parameters for the PHOS Signal Model

As shown in Figure 4.3 the time dependent signal sits on a baseline and in order to take this into account Equation 4.13 could be rewritten as

$$f(t) = A^2_{adc}\big(\frac{t-t_0}{\tau_{shaper}}\big)^2 \exp\big(-2\frac{t-t_0}{\tau_{shaper}}\big) + b \tag{G.4}$$

where b denotes the baseline. Equation G.5 has four parameters: A, t_o, τ, and b. The question is which one of these parameters can be considered to be "free". Choosing different combinations of free parameters yields different signal models, and consequently different mvlbs. In general choosing too many free parameters deteriorates the variance of the estimate of the individual parameters. One should avoid "over-modeling" of the signal and restrict the number of free parameters to a minimum [62]. The two parameters that is certainly known to vary from event to event is the amplitude A and the peak position.

As shown in Figure E.1 it is shown that the baseline is approximately constant from event to event. In addition it was observed during the cosmic data taking with the first PHOS module in 2006 that the long term stability of the baseline was very good[2], there is therefore no reason to use the baseline as a free parameter in the model. Furthermore, as outlined in Section 4.7.3, the baseline will be subtracted by the configuration of the electronics and subsequently by the subtraction of baseline residuals during reconstruction so that the baseline can be assumed to be zero prior to the evaluation of the amplitude and the peak position.

The parameter τ is determined by the actual values of the resistors and the capacitors on the shaper circuits of the FEECs, It is therefore not reasonable to assume that the true value of τ fluctuates from event to event for a given channel.

To summarize: although Equation G.5 has four parameters only two of them, A and t_0 can be considered to be free and the baseline is assumed to be zero. Equation G.5 can then be written as

$$f(t) = A^2\big(\frac{t-t_0}{\tau^{const}_{shaper}}\big)^2 \exp\big(-2\frac{t-t_0}{\tau^{const}_{shaper}}\big) \tag{G.5}$$

Where τ^{const}_{shaper} and is constant from event to event for an individual readout channel whereas this constant values might differ from channel to channel. In order to simplify the notation we shall set $\tau^{const}_{shaper} = \tau$ from here on.

A realization $\boldsymbol{m} \in \boldsymbol{X}$ for PHOS is the sequence of time dependent signal after sampling by the ALTRO ADC indicated bu the white dots in Figure G.1. The red line indicates the deterministic output of LSI1 and in the sampling indexes corresponds to the expectation value $E\{X_n\}$ of sample m_n. The random contribution in sample n is obtained by the difference $m_n - E\{m_n\} = \epsilon_n$. A random process with zero mean is obtain by subtracting

[2]The same baseline matrix to subtract the baseline was used over a period of two months.

the deterministic G2 function from the measurement to obtain the errors i.e

$$\boldsymbol{m} - E\{\boldsymbol{m}\} = \boldsymbol{\epsilon} \tag{G.6}$$

The statistical properties of $\boldsymbol{\theta}$ and therefor also $\boldsymbol{\epsilon} = [\epsilon_0, \epsilon_1, \cdots, \epsilon_{N-1}]$ is given by the covariance matrix. In this context it is useful to add a error term to Equation G.5 to obtain. G.5 can then be written as

$$f(t)_a^{rand} = A^2(\frac{t-t_0}{\tau})^2 \exp(-2\frac{t-t_0}{\tau}) + \epsilon(t) \tag{G.7}$$

So that $f(t)_a^{rand}$ becomes a continuous random variable that is a function of the real time t. Furthermore we shall distinguish between the analogue continuous signal denoted $f(t)_a^{rand}$ and the digitized one denoted $F(n)_d^{rand}$ where the two signals are related by the

$$F(n)_d^{rand} = f(t = nT_s)_a^{rand} \tag{G.8}$$

T_s is the sampling interval given by the inverse of the sampling frequency i.e.

$$T_s = 1/f_s \tag{G.9}$$

An ALTRO ADC sampling frequency of 10 MHz thus gives a sampling interval of

$$T_s = \frac{1}{10^7 s^{-1}} = 100ns. \tag{G.10}$$

The result of the digitization is to obtain the set of discrete random variable $\boldsymbol{X} = [X_0, X_1, \cdots, X_{N-1}]$ from the continuous random variable given by Equation G.7. where N denotes the number of samples. In addition we have

$$X_i \in [0, 1, \cdots, 1023] \qquad \text{for each} X_i \in \boldsymbol{X} \tag{G.11}$$

since the ALTRO ADC is 10 bit.

During evaluation of energy and TOF from lets say a realization $\boldsymbol{m} \in \boldsymbol{X}$ one always assumes the TOF to be in entities of T_s (i.e in entities of sample indexes) and the energy in entities of ADC levels. The software model used is thus on the form

$$F(n) = A_{adc}^2\big(\frac{n-\Delta_{t_0}}{\Delta_\tau}\big)^2 \exp\big(-2\frac{n-\Delta_{t_0}}{\Delta_\tau}\big) \tag{G.12}$$

where $\Delta_\tau = \Delta_\tau(fs)$ is a constant which is a function of the sampling frequency. For $f_s = 10$ MHz and a rise-time of $\tau = 2\mu s$ we have

$$\tau = 20 \cdot T_s = \Delta_\tau \cdot T_s. \tag{G.13}$$

The software model in this case becomes

$$F(n) = A_{adc}^2\big(\frac{n-\Delta_{t_0}}{20}\big)^2 \exp\big(-2\frac{n-\Delta_{t_0}}{20}\big) \tag{G.14}$$

where n runs over the number of samples, and A_{adc} and Δ_{t_0} are the two free parameters of the model.

The real time is given by the relation

$$t_0 = \Delta_{t_0} \cdot T_s \tag{G.15}$$

and the real energy deposited in channel i is given by the relation

$$E_i = \alpha_i \cdot A_{adc}^i \tag{G.16}$$

where A_{adc}^i is the ADC amplitude measured in channel i and α_i is the channel specific amplitude to energy conversion factor. This conversion factor is also denoted a *calibration constant* since the fine tuning of calibration constants for all the channels is a part of the calibration procedure for PHOS (see Section 7.3).

G.1.3 Desirable Properties of an Estimator

(i) *Consistency.* The estimator should converge to the true values of the parameters as the number of observations increases. By definition an estimator is unbiased if given any $\varepsilon > 0$ and any $\eta > 0$ there exists an N such that

$$P(|\tilde{\theta}_N - \theta_t| > \varepsilon) < \eta \tag{G.17}$$

where $\tilde{\theta}_N$ estimate based on N observations for all N > N. Here P denotes the probability.
(ii) *Unbiasedness.* An estimator is unbiased if the expectation value of $\tilde{\theta}_n$ is identical to the true value θ_t or

$$E\{\tilde{\theta}\} = \theta_t \tag{G.18}$$

(iii) *Minimum Variance.* The variance $Var(\tilde{\theta})$ should be as small as possible under the constraint that the estimator is unbiased.
(iv) *Robustness* The estimator should be as insensitive as possible to any discrepancy between the real and the assumed probability distribution of individual measurements.

It is useful for the subsequent discussion to introduce the following terms: *sufficient statistics*, *the Fisher information*, and *minimum variance lower bound*. The exact definition can be found in for instance [63].

G.1.3.1 The Fisher Information

The requirements given here are the ones that leads to the definition of *information* as defined by R.A Fisher [63]. The requirements are: 1) The information should increase with the number of measurements, doubling the amount of measurements should double the information if the measurements are independent. 2) The information should be dependent on what one wants to measure. Data that is irrelevant for the observables should not increase the information. 3) The information must be related to the precision. The larger the information, the better the precision of the experiment.

Let $\boldsymbol{X} = [X_0, X_1, \cdots, X_{N-1}]$ be a random process described by a model having the parameters $\boldsymbol{\theta} = [\theta_0, \theta_1, \cdots, \theta_{k-1}]$. The joint probability density function of $\boldsymbol{X}$ is denoted $P(\boldsymbol{X}|\boldsymbol{\theta})$.

Now let $\boldsymbol{m} = [m_0, m_1, \cdots, m_{N-1}]$ be a realization of $\boldsymbol{X}$. The *likelihood function* is defined as

$$L(\boldsymbol{\theta}) = P(\boldsymbol{m}|\boldsymbol{\theta}). \tag{G.19}$$

L($\boldsymbol{\theta}$) is the *likelihood* of a set of model parameters $\boldsymbol{\theta}$ and is a function of $\boldsymbol{\theta}$ only.

If $\boldsymbol{\theta}$ has k parameters the *information* is a $k \times k$ matrix and element i,j of this matrix is defined as

$$[I_m(\boldsymbol{\theta})]_{i,j} = \int \left\{ \left[\frac{\partial \ln L(\boldsymbol{\theta})}{\partial \theta_i}\bigg|_{\boldsymbol{\theta_t}} \Big/ L(\boldsymbol{\theta_t}) \right] \times \left[\frac{\partial \ln L(\boldsymbol{\theta})}{\partial \theta_j}\bigg|_{\boldsymbol{\theta_t}} \Big/ L(\boldsymbol{\theta_t}) \right] \right\} \times L(\boldsymbol{\theta}) d\boldsymbol{m} \tag{G.20}$$

If the measurements are uncorrelated $L(\boldsymbol{\theta})$ factorizes and and the total information is just the sum of the information of the individual measurements.

G.1.3.2 The Cramer-Rao Inequality

The variance of an unbiased estimate is limited downwards by the Cramer-Rao inequality which is the inverse of the Fisher information

$$\mathrm{Var}(\theta_i) \geq [I_m(\boldsymbol{\theta})]^{-1}_{i,i}. \tag{G.21}$$

If equality holds in Equation the estimator is said to be *minimum variance lower bound* (mvlb). An mvlb estimator is thus the best possible one in terms of variance.

G.1.3.3 Sufficient Statistics

The statistics is sufficient if enough measurements are already available so that the presence of additional measurements does not not affect $Var(\tilde{\boldsymbol{\theta}})$. For PHOS one can ensure that sufficient statistics is obtained by using enough samples to cover the whole signal (typically 130 - 150 samples is sufficient). Furthermore the sampling frequency should be high enough for the Nyquist criterion to be satisfied. Referring to Figure 4.6 the -20 db cutoff frequency is ∼1MHz so that the ADC sampling rate should be at least 2MHz.

Most[3] of the data taken with PHOS in the period 2003 - 2008 was actually taken taken with a sampling rate of 10 MHz. One of the reason was that in the previous version of the RCU firmware the ADC clock signals to the ADCs was fixed to 10 MHz. In the most recent version of the firmware this is changed and it will be possible to use different clock frequency of the ADC sampling clock. Another reason is that in presence of problems such as noise it is useful to over-sample because the resolution in frequency spectrum obtained during Fourier analysis improves. The fact that the PHOS detector signals are oversampled gives an interesting opportunity to demonstrate the meaning of sufficient statistics in practice but before doing that we shall discuss the next term in the list.

Example: Sufficient Statistics from PHOS Raw Data The dependence of the TOF resolution on the ALTRO ADC sampling frequency was investigated. The results are discussed further in Appendix H. A plot of the results given in Table H.1 is shown in Figure G.2. The red circles indicates the measurements. Data was taken with a 10 MHz sampling frequency. Different sampling frequencies were emulated by discarding data during the analysis by using only every second, every third etc sample during the analysis. As shown 4.6 the -20 db cutoff frequency of the shaper is $\sim$1 MHz. One should therefore expect[4] that the measurements/samples become correlated when the sampling rate is higher than ~ 2 MHz. For uncorrelated measurements/samples the TOF resolution is related to to the sampling frequency f_s[43] by

$$\sigma_{TOF}(f_s) = A/\sqrt{f_s/Hz}ns \tag{G.22}$$

The data is therefore fitted with this function in the range 0.4 - 2 MHz giving $\sigma_{TOF}(f_s) = 2.79/\sqrt{f_s/Hz}ns$. The blue dotted line shows the extrapolation to the range 2 - 10 MHz. This is the resolution expected if the measurements was uncorrelated also for $f_s > 2$ MHz. As seen in the figure the σ_{TOF} flattens out sharply above 2 MHz. Doubling f_s from lets say 2 to 4 MHz does not significantly increase the information because the additional measurements are highly correlated with the measurements obtained at 2 MHz. They could have been predicted from the existing ones i.e the information about those additional samples was already present at 2 MHz. To summarize: Sampling at 2 MHz provides sufficient statistics for σ_{TOF}

G.1.3.4 Efficiency of an Estimator

The *efficiency* is defined as the ratio between $Var(\tilde{\theta}$ and the mvlb. If the ration is one the estimator is also said to be *fully efficient*. The two terms *mvlb* and *fully efficient* describe

[3]Some data was also taken with a sampling rate of 20 MHz, but this required a special version of the RCU firmware.

[4]A direct measurement of the correlation is not possible since the sampling clock and the trigger was not synchronized when this data was taken.

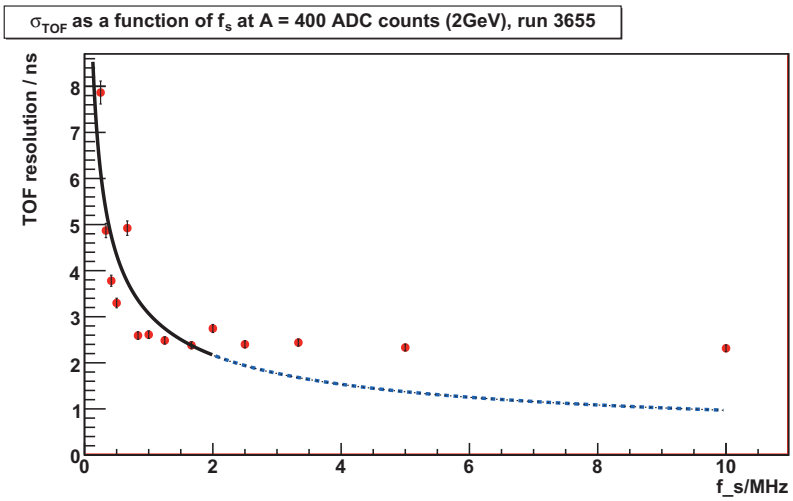


Figure G.2: The TOF resolution as a function of the *emulated* sampling rate $f_s^{emulated}$. The red filled circles are the measurements. The solid black line is the result of fitting with the function $A/\sqrt{f_s}$ in the range 0 -2 MHz giving $\sigma_{TOF}(f_s^{emulated}) = 2.79/\sqrt{f_s^{emulated}/Hz}ns$. The dotted blue line is the extrapolation of the fit results to the range 2 -10 MHz.

the same property and will be used interchangeably in the rest of this chapter. Whereas *fully efficient* gives a lower bound of the variance of any estimator the therm *optimum* is used for an estimator that is the best among a class of estimators. In the rest of this text only linear estimators will be considered and we will make the following definition.

Definition 1 *An estimator is* optimum *if it is the best in the class of linear estimators, assuming a given model.*

It is important to point out that the mvlb is model dependent whereas the algorithm used to obtain the estimator is not. This has the implication that it is possible that different algorithms can produce a fully efficient estimator. It will the be demonstrated in Section G.1.4 and G.1.5 that both the Least Mean Square(LMS) and the *Peakfinder* can be used to obtain fully efficient estimates of the amplitude and peak position using Equation 4.13 as a model.

G.1.3.5 The Covariance Matrix

Before defining the covariance matrix it is useful to compare the space spanned by a set of random variables with the space spanned by a set of deterministic basis vectors. The

Deterministic case	Random case
Orthogonal with unit length $e_i \cdot e_j = 0; \quad i \neq j$	Uncorrelated with unit variance $E\{X_i X_j\} = 0; \quad i \neq j$
Orthogonal with different lengths	Uncorrelated but with different variances
Non orthogonal with different lengths	Correlated with different variances

Table G.1: Comparison between basis vectors spanning a deterministic and a random space.

statistical properties of $\boldsymbol{X}$ is given by the covariance matrix defined as

$$\mathbf{C}_X = E\{(\boldsymbol{X} - E\{\boldsymbol{X}\})(\boldsymbol{X} - E\{\boldsymbol{X}\})^H\}. \tag{G.23}$$

Here E denotes the expectation value and H the Hermitan transpose. Writing out the outer product given by Equation G.23 on matrix form defining the covariance

$$E\{(X_i - E\{X_i\}) \cdot (X_j - E\{X_j\})\} \equiv Cov(X_i, X_j) \equiv c_{X_i,X_j} \tag{G.24}$$

and the variance

$$Cov(X_i, X_i) \equiv Var(X_i) = \sigma^2_{X_i}$$

yields, for a random process with N variables.

$$\mathbf{C}_X = \begin{bmatrix} \sigma^2_{X_0} & c_{X_1,X_0} & \cdots & c_{X_{N-1},X_0} \\ c_{X_0,X_1} & \sigma^2_{X_1} & \cdots & c_{X_{N-1},X_1} \\ c_{X_0,X_2} & c_{X_1,X_2} & \cdots & \\ \vdots & \vdots & \ddots & \\ c_{X_0,X_{N-1}} & c_{X_1,X_{1,N-1}} & \cdots & \sigma^2_{X_{N-1}} \end{bmatrix} \tag{G.25}$$

The diagonal of the matrix represents the *first moments* and the off diagonal elements the *second moment* of the random process $\boldsymbol{X}$. If the process is a Gaussian then everything that can be known about the process is represented in the matrix since the statistical properties of Gaussian processes only depends on the first and second order moment. In the next two sections two estimators will be considered: The Least Mean Square (LMS) and the Peak Finder (PF) algorithm.

G.1.4 The Least Mean Square Estimator

The Least Mean Square (LMS) method is one of the most popular methods for extraction of model parameters from a set of measurements. One of the reasons is that this method gives fully efficient estimates of the model parameters if the errors are Gaussian and i.i.e (identical and independent errors). The M = *Mean* in the LMS denotes the statistical

expectation value. The aim of the LMS is to find the set of model parameters $\tilde{\boldsymbol{\theta}}$ that minimizes the expectation value of the distance between a measurement $\boldsymbol{m}$ and the constraint surface defined in Section G.1.

Figure G.3 shows the constraint surface for the G2 function in the range $t_0 \in [1, \cdots, 3]\mu s$ and $A \in [0, \cdots, 100]$ in arbitrary units [5] assuming two free parameters A and t_0. The surface is then defined by all possible combinations of A and t_0. The three dimensional space is spanned by the three random variables X_0, X_1 and X_2 denoted x[0], x[1] and x[2] in the figure. The constraint surface is a two dimensional "parameter" space which is a subspace of the three dimensional measurement space indicated by the surface. The reason for the large range of $X[0]$ compared to $X[1]$ and $X[2]$ is that for $t_0 \in [1, \cdots, 3]\mu s$ $X[0]$ obtains values over the peak of the signal while $X[1]$ and $X[2]$ falls off exponentially along the tail.

Figure **??** shows a simulated G2 signal with the addition of a Gaussian noise which magnitude is indicated by the error bars. As seen in the figure the fit is not perfect. A non perfect fit corresponds to the situation when the set of measurements does not lie in the constraint surface. This is typically the case. It is indicated in Figure G.3: A measurement $\boldsymbol{m}$ lies outside the constraint surface. The minimization of the distance between the measuremnt and the surface can only be done correctly if the basis vectors spanning the measurement space, i.e. the covariance matrix, is known. In order to obtain the correct minimum it is necessary to know the variance of the measurements and how the measurements are correlated. The measurements has to be weighted according to their variance and covariance with other measurements. This weights are provided by the weight matrix which is the inverse of the covariance matrix.

An optimum estimates close to fully efficient can be obtained for the LMS if the model can be reasonably well be approximated by a linear model in the neighbourhood of the measurements. If the model is non linear, such as Equation 4.11, the model can be linearized by a first order Taylor expansion around an initial guess $\boldsymbol{\theta_0}$ of the model parameters. In the case of the PHOS shaper output we shall define

$$\boldsymbol{\theta_0} = [A^0, t_0^0] \tag{G.26}$$

where A^0 and t^0 are the initial guesses of the amplitude and peak position respectively. The linear approximation $\boldsymbol{\theta}$ from the expansion point $\boldsymbol{\theta_0}$ is

$$f(\boldsymbol{\theta}) = f(\boldsymbol{\theta_0}) + \mathbf{A} \cdot (\boldsymbol{\theta} - \boldsymbol{\theta_0}) \tag{G.27}$$

where

$$\mathbf{A} = \frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \qquad \text{at } \boldsymbol{\theta} = \boldsymbol{\theta_0} \tag{G.28}$$

[5] The amplitude could have been given in terms of ADC counts but since the amplitude is not relevant here the subscript is omitted.

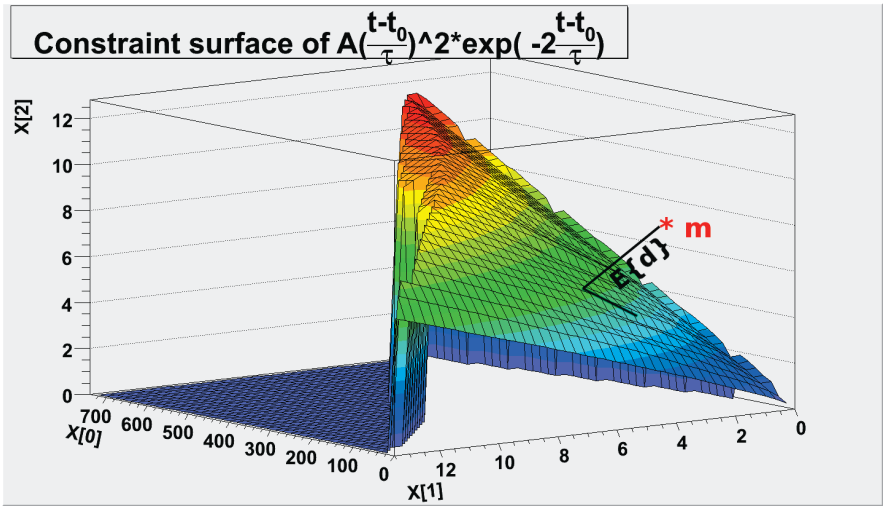


Figure G.3: Constraint surface of the gamma-2 function assuming two observables, A and t_0 and three measurements/samples. A measurement m lies in general not on the constraint surface. The objective of the LMS is to minimize the expectation value E{d} between the measurement and the surface. This minimum expectation value defines the *normal* between m and the surface. The parameter estimates are given by the intersection between this *normal* and the surface.

The *weight* matrix $\mathbf{W}$ is defined as inverse of the covariance matrix

$$\mathbf{W} = {\mathbf{C_x}}^{-1} \tag{G.29}$$

The LMS minimizes the function

$$M = [f(\boldsymbol{\theta_0}) + A \cdot (\boldsymbol{\theta} - \boldsymbol{\theta_0}) - \boldsymbol{m}]^T \mathbf{W} [f(\boldsymbol{\theta_0}) + A \cdot (\boldsymbol{\theta} - \boldsymbol{\theta_0}) - \boldsymbol{m}] \tag{G.30}$$

The way many nonlinear fitting algorithms work is to use Equation G.27 to make a new guess, lets say ${\boldsymbol{\theta_0}}^{0+1}$ for instance using Newtons method, then use this new guess to make a new one and so on. This iterative procedure continues until the minimum of Equation G.30 is known to a given (user defined) precision constraint put on the derivative. That is, if the first derivative is closer to zero than the constraint the estimate is said to have converged. Several LMS algorithms are available. One of the most popular ones is the Levenberg Marquardt algorithm [64]. In general a trade-of has to be made between the convergence speed and the precision obtained in the parameter. A better precision implies a slower convergence an therefor more iterations.

After for instance k iterations one has estimate θ^{0+k} that satisfies the precision requirement.

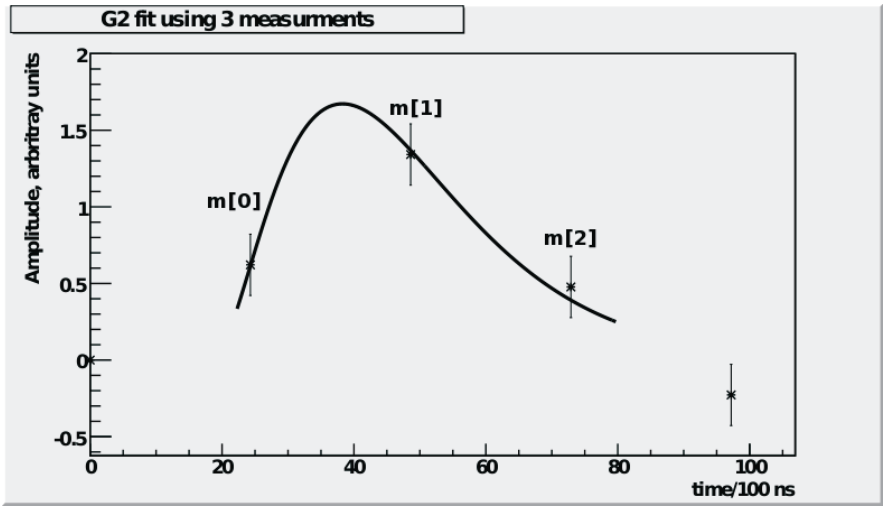


Figure G.4: A simulated signal, with random Gaussian noise added and then fitted with a G2 function.

In the simplest case the errors are i.i.e and the weight matrix reduces to

$$\mathbf{W} = \frac{1}{\sigma^2} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{G.31}$$

This is the default used by most algorithms. If the errors are independent (uncorrelated) but not identical then the weight matrix becomes

$$\mathbf{W} = \begin{bmatrix} 1/\sigma^2_{x(0)} & 0 & \dots & 0 \\ 0 & 1/\sigma^2_{x(1)} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & 1/\sigma^2_{x(N-1)} \end{bmatrix} \tag{G.32}$$

It is readily seen from Equation G.32 why it is called *weights*. It cancel the error of the square magnitude of $\boldsymbol{m}$ on each side of the weight matrix. In the case when the covariance matrix is not diagonal the situation becomes more complicated and one generally has to solve Equation G.29 numerically.

	Errors	
Model	Non-Gaussian	Gaussian
Linear	Optimal	Fully efficient for a finite sample with sufficient statistics
Nonlinear	No general properties for a finite sample, but asymptotically fully efficient	Equivalent to the Maximum likelihood estimate for a finite sample, and therefore asymptotically fully efficient

Table G.2: Properties of the LMS.

The efficiency of the LMS depends on whether or not the random variable $\boldsymbol{X}$ is Gaussian and whether or not the signal model is linear. The properties of the LMS as a function of these properties is given in Table G.2[65]

G.1.5 The Peakfinder algorithm

G.1.5.1 Optimum Parameter Estimation under the Constraints given by the PF Algorithm

It is possible find the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by calculating the least square solutions of the overdetermined systems given by Equation 6.7 and 6.10. This will however not give the optimum coefficients in the presence of correlated noise and random time-delays. In order to obtain the optimum coefficients it is necessary to treat the peak-finder method within the framework of statistical signal processing.

Since the timing and energy will be reconstructed from a digitized signal we shall consider the detector signals after ADC quantization. The effect of ADC quantization is a complicated issue that is pending further investigation. Quantization effects will have to be measured during the calibration of the detector, possibly on a channel by channel basis.

Optimum Coefficients for Amplitude Let $\hat{A}_{m\varepsilon}$ be the estimated amplitude at time-delay $m\varepsilon$. The optimum coefficients are found by minimizing the sum of squares of the errors $\epsilon(m) = A - \hat{A}_{m\varepsilon}$ with respect to the filter coefficients $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \cdots, \alpha_{N-1}]$, that is, to minimize the cost function

$$\xi = E\{\epsilon(m)\epsilon(m)^*\}$$

with respect to the coefficients $\boldsymbol{\alpha}$ i.e. to find a set of coefficients such that

$$E\left\{\epsilon(m)\frac{\partial\epsilon^*(m)}{\partial\alpha_l^*}\right\} = 0, \qquad \begin{array}{l} l = 0, 1, \cdots N-1. \\ m = 0, 1, \cdots M-1 \end{array}$$

with the substitution

$$\epsilon^*(m) = A - \hat{A}_m = A - \sum_{k=0}^{N-1} \alpha_k^* g_A^*(k)_{m\varepsilon}$$

and

$$\frac{\partial}{\partial \alpha_l^*} \left\{ \sum_{k=0}^{N-1} \alpha_k^* g_A^*(k)_{m\varepsilon} \right\} = g_A^*(k)_{m\varepsilon}$$

the optimization problem can be written

$$E\left\{ \sum_{k=0}^{N-1} \alpha_k g_A(k)_{m\varepsilon} g_A^*(l)_{m\varepsilon} - A \cdot g_A^*(l)_{m\varepsilon} \right\} = 0$$
$$l = 0, 1 \cdots N-1$$
$$m = 0, 1 \cdots M-1$$

Here we assume that

$$g_A(n) \approx A \cdot \hat{y}_1(n) + v(n).$$

and define

$$\begin{aligned} E\{A \cdot g^*(l)_{m\varepsilon}\} &= A \cdot E\{y(l)_{m\varepsilon}\} + \underbrace{E\{v(n)\}}_{=0} \\ &= A \cdot \overline{y}(l) \end{aligned}$$

Here $\overline{y}(l)$ denotes the average of $y(l)_\varsigma$ over all possible time-delays. This means that $\overline{y}(l)$ depends on the distribution of time delays. During previous beam-tests of the 256 prototype PHOS module the sampling clock was not synchronized with the trigger so that the time delays were uniformly distributed within one sampling period T_s. In this case $\overline{y}(l)$ is approximately given by

$$\overline{y}(l) \approx s(T_s \cdot l) + \frac{T_s}{2} \cdot \frac{d}{dt}[s(t)]_{t=lT_s}$$

$\overline{y}(l)$ will most likely have to be determined by measurements, for instance by analyzing pulser data. In addition we define the auto-correlation of g as

$$\begin{aligned} r_g^A(n,l) &\equiv E\{g_A(n+l)_{m\varepsilon} g_A^*(n)_{m\varepsilon}\} \\ &= E\{[y_A(n+l)_{m\varepsilon} + v(n+l)][y_A^*(n)_{m\varepsilon} + v^*(n)]\} \\ &= E\{y_A(n+l)_{m\varepsilon} y_A^*(n)_{m\varepsilon}\} + \underbrace{E\{y_A(n+l)_{m\varepsilon} v^*(n)\}}_{=0} \\ &+ \underbrace{E\{v(n+l) y_A^*(n)_{m\varepsilon}\}}_{=0} + E\{v(n+l)v^*(n)\} \\ &= E\{y_A(n+l)_{m\varepsilon} y_A^*(n)_{m\varepsilon}\} + E\{v(n+l)v^*(n)\}. \end{aligned}$$

We define the *deterministic* auto-correlation of y as

$$r_y^A(n,l) \equiv E\{y_A(n+l)_{m\varepsilon} y_A^*(n)_{m\varepsilon}\} \tag{G.33}$$

As seen from Equation G.33 $r_y(n,l)$ depends on the start-index l, the time difference $l-n$ and the distribution of time delays $\varsigma = m\varepsilon$ with respect to the ADC sampling clock. For the noise term the auto-correlation only depends on the separation in time and we shall define it as

$$r_v(l) \equiv E\{v(n+l)v^*(n)\}.$$

For a signal with amplitude A it is assumed that the auto-correlation can be approximated as

$$r_g^A(n,l) = A \cdot \hat{r}_y(n,l) + r_v(l)$$

In analysis one will use a sub-array of the digitized signal that starts at lets say i samples into the pulse. Equation G.34 can be expressed in matrix form as

$$\begin{bmatrix} r_g^A(i,0) & r_g^A(i,1) & \cdots & r_g^A(i,N-1) \\ r_g^{A*}(i,1) & r_g^A(i,0) & \cdots & r_g^A(i,N-2) \\ r_g^{A*}(i,2) & r_g^{A*}(i,1) & \cdots & r_g^{A*}(i,N-3) \\ \vdots & \vdots & \vdots & \\ r_g^{A*}(i,N-1) & r_g^{A*}(i,N-2) & \cdots & r_g^A(i,0) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} = A \begin{bmatrix} \overline{y}_A(i) \\ \overline{y}_A(i+1) \\ \overline{y}_A(i+2) \\ \vdots \\ \overline{y}_A(i+N-1) \end{bmatrix}$$

For large amplitudes where $y_A(n) >> v(n)$ and $r_g^A(n,l) \approx r_y^A(n,l)$. This means that for large amplitudes the optimal coefficients does not depend so much on whether or not the errors are correlated. In the limit of a large amplitude we assume that.

$$y(l)_A \approx A\overline{\hat{y}(l)}$$

and

$$r_g^A(n,l) \approx A^2 \hat{r}(n.l)$$

Here $\hat{r}_(n,l)$ denotes the auto-correlation of the deterministic part of the unit amplitude signal before ADC quantization while $\overline{\hat{y}(l)}$ denotes the average over different time delays for a unit amplitude signal before ADC quantization.

In the limit of large amplitudes we then get a factor A^2 on both sides of equation G.1.5.1 that cancels so that Equation G.1.5.1 can be rewritten as.

$$\begin{bmatrix} \hat{r}_g(i,0) & \hat{r}_g(i,1) & \cdots & \hat{r}_g(i,N-1) \\ \hat{r}_g^*(i,1) & \hat{r}_g(i,0) & \cdots & \hat{r}_g(i,N-2) \\ \hat{r}_g^*(i,2) & \hat{r}_g^*(i,1) & \cdots & \hat{r}_g^*(i,N-3) \\ \vdots & \vdots & \vdots & \\ \hat{r}_g^*(i,N-1) & \hat{r}_g^*(i,N-2) & \cdots & \hat{r}_g(i,0) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N-1} \end{bmatrix} = \begin{bmatrix} \overline{\hat{y}}(i) \\ \overline{\hat{y}}(i+1) \\ \overline{\hat{y}}(i+2) \\ \vdots \\ \overline{\hat{y}}(i+N-1) \end{bmatrix}$$

This can be written more compact as

$$\mathbf{R}_g \cdot \boldsymbol{\alpha} = \boldsymbol{y}$$
$$\boldsymbol{\alpha} = \mathbf{R}_g^{-1}\boldsymbol{y}$$

Optimum Coefficients for Timing The optimum coefficients for timing is found in a similar way as for energy except that the amplitude A_m is replaced by the time-delay $\varsigma = m\varepsilon$. As stated in section 6.2.1.2 the signal must be normalized to unit amplitude in order to yield the correct result and we assume that

$$g_1(n) \approx \frac{1}{\hat{A}} g_A(n).$$

In addition the average value of the time delays is defined as

$$E\{\varsigma\} \equiv \overline{\varsigma}$$

The optimization problem can then be expressed as

$$E\left\{\varsigma - \sum_{k=0}^{N-1} \beta_l g_1(k) g_1^*(l)\right\} = 0$$
$$\Updownarrow$$
$$E\left\{\sum_{k=0}^{N-1} \beta_l g_1(k) g_1^*(l)\right\} = \overline{\varsigma}$$
$$l = 0, 1 \cdots N-1$$
$$m = 0, 1 \cdots M-1$$

With the auto-correlation defined the same way as before this can be expressed in matrix form as

$$\begin{bmatrix} r_g(i,0) & r_g(i,1) & \cdots & r_g(i,N-1) \\ r_g^*(i,1) & r_g(i,0) & \cdots & r_g(i,N-2) \\ r_g^*(i,2) & r_g^*(i,1) & \cdots & r_g^*(i,N-3) \\ \vdots & \vdots & \vdots & \\ r_g^*(i,N-1) & r_g^*(i,N-2) & \cdots & r_g(i,0) \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{N-1} \end{bmatrix} = \overline{\varsigma} \begin{bmatrix} g(i,0) \\ g(i,1) \\ g(i,2) \\ \vdots \\ g(i,N-1) \end{bmatrix}$$

This can be written more compact as

$$\mathbf{R_g}\boldsymbol{\beta} = \overline{\boldsymbol{\varsigma}}\mathbf{g}$$
$$\boldsymbol{\beta} = \mathbf{R_g^{-1}}\overline{\boldsymbol{\varsigma}}\mathbf{g}$$

G.1.6 Discussion

In order to obtain correct results from a fitting procedure the signal model must be correct and the weight matrix, and therefore the covariance matrix, must be known as exactly as possible.

G.1.6.1 The LMS Estimator

Most fitting algorithm requires an *initial guess* of the parameters in order to converge to an optimum estimate $\tilde{\boldsymbol{\theta}}$.In general, the better the initial guess, the faster the algorithm will converge. If this initial guess is too far from the actual values the algorithm might diverge. An initial guess can be obtain for instance by pre-processing of the data. For the PHOS the initial guess for the amplitude can be taken to be the max signal bin value and the TOF the time bin of the max signal bin subtracted the rise-time (τ) and multiplied by T_s. Most fitting algorithms assumes by default that the weight matrix is the identity matrix[6] and if it is something else then this information has to be supplied by the user. For PHOS in particular the noise is typically higher for samples in the falling edge than in the rising edge of the signal however since the covariance matrix was not measured so far[7]. As a consequence it was observed that including too many samples of the tail when doing the fit actually worsened the variance of $\tilde{\boldsymbol{\theta}}$. It was found that under the (wrong) assumption that the weight matrix is the identity matrix the optimum number of samples to use was ∼90 samples whereas using the 130 - 150 samples[8] needed to cover the whole signal only worsened the variance of the estimates.

If the signal model is incorrect the estimates might be biased. As discussed in Chapter 4 the real signal cannot be expected to be exactly a G2 signal. Figure G.5 shows a typical example of the effect of a small discrepancy between real PHOS signal and an ideal a G2 signal. The black filled area shows the actual signal. The red line shows the result of a G2 fit in the time bin range 10 - 100. The blue line shows a G2 fit in the range 30 - 40. The fit indicated by the blue line is used here only to indicate the systematic error introduced by using a wrong model. As seen in the figure the result from the fitting says that the amplitude is $\sim$ 715 ADC counts however as seen in the upper right corner of the figure the actual amplitude is around $\sim$ 732 ADC counts giving a systematic error of $\sim 2\%$. For estimate of t_0 the value is shifted by $\sim$ 1.5 time bins corresponding to 150 ns. Given that one wants to obtain an inter-calibration of better than 0.1% for energy and TOF measurements better than at $\sim$ 1 ns at 2GeV, it is clear that these systematic

[6]The real noise is of course rarely of unit variance but as seen from seen from Equation G.30 the weight matrix factors out of the equation in the special case when the covariance matrix is the identify matrix multiplied by a constant. It therefore does not matter for the evaluation of $\tilde{\boldsymbol{\theta}}$ if the weight matrix is the identity matrix or the identity matrix multiplied by a constant.

[7]To measure the covariance matrix would require real data to be taken with trigger synchronized with the ADC sampling clock. This data can only be taken once LHC makes collisions.

[8]Depending on the amplitude

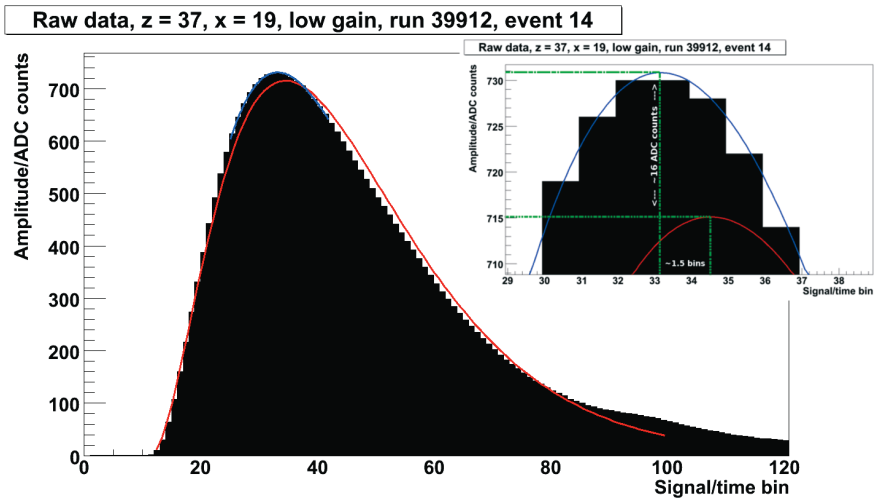


Figure G.5: Event from electron beam in 2006

errors has to be corrected for.

If the discrepancy between the assumed model and the actual on is not to large an biased estimator can be made unbiased by adding or multiplying the estimates with a constant. If one assumes for PHOS that the shape of the signal is approximately independent of the amplitude then the systematic error of the TOF estimate should be a constant independent of the amplitude whereas the systematic error in the estimate of the amplitude should scale with the amplitude. That is

$$t_0^{unbiased} = t_0^{biased} + t_0^{shift} \tag{G.34}$$

and

$$A^{unbiased} = k \cdot A^{biased} \tag{G.35}$$

Referring to Figure G.5 t_0^{shift} and k would in this case be $\sim$ 150 ns and $\sim$1.02.

G.1.6.2 The Peak-Finder Estimator

When using the Peak-Finder algorithm the weights of the measurements are choosen in order to minimize the systematic errors. The wiights is "used up" for this purphose and it will not be possible to find weights that both minimizes the variance and the systematic error. The proof is given by concidering Equation 6.7 wich in general has a solution

$$\boldsymbol{\beta} \neq [1, 1, \cdots, 1] \tag{G.36}$$

with the specila case when the wight matrix is the identity matrix. On the other hand this same property makes the peak finder algortihm robust agains measurements with an unexpected statistical property.

G.2 Discussion

For large signal amplitudes the deterministic part of the auto-correlation matrix dominates over the noise part. Therefore the PF coefficients is not affected by correlated noise for large amplitudes. For the same reason the optimum coefficients does not depend on amplitude. Exactly how high the amplitude must be depends on the actual noise which has to be measured.

For low amplitudes it might be necessary to use different sets of coefficients for different ranges of amplitudes. This is an issue that must be investigated by simulations and analysis of real data.

G.2.1 The Systematic Error of the Peak-Finder Algorithm

In general the PF algorithm will induce some systematic errors in the estimates of TOF and Amplitude because the overdetermined set of Equations given by 6.9 and 6.5 cannot be solved exactly. However a sufficient condition for the overdetermined system of equations to be exactly solvable is that the representative curve is a line in the space of the measurements. The representative curve will be span a line when.

$$\frac{\partial y}{\partial x}_{x=t_0} x \in \{0, ...T_s\} \approx const$$

G.2.2 Propagation of errors

In order to compare the PF with the LMS we shall look at the propagation of errors of the two methods. Only the error propagation for timing is considered but a similar argument will hold also for energy.

Propagation of Errors for the LMS For a least square fit to a correct model we start by assuming the following.

1. $f(t) = s(t) + e(t)$ where $s(t)$ is the (deterministic) signal and $e(t)$ is the noise.
2. Perform least square fit and assume $e(n) = g(n) - s(nT_s + t_0)$
3. $t_0^k = s^{-1}\hat{(}kT_s - t_0)$
4. $\mathrm{Var}[e(n)] = constant = \sigma$

5. $\Delta f_i = \left.\frac{dt}{dg}\right|_{t=iT_s-t_0}$

For a total of N measurements, the best estimate of t_0 and its standard error is

$$\hat{t}_0 = \underbrace{\frac{\sum_{i=0}^{N-1} 1/\Delta f_i \cdot t_0^i}{\sum_{i=0}^{N-1}(1/\Delta f_i)^2}}_{\text{best estimate of } t_0} \pm \underbrace{\sqrt{\frac{\sigma^2 + 2\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}\frac{\mathrm{Cov}(e(n),e(m))\delta_{n,m}}{\Delta f_n \Delta f_m}}{\sum_{i=0}^{N-1}(1/\Delta f_i)^2}}}_{\text{standard error of estimate}}. \tag{G.37}$$

Next we will consider the special case when errors are uncorrelated. Then the covariance term of equation G.37 vanishes. In order to simplify the notation we also define

$$k_i \equiv \frac{1/\Delta f_i}{\sum_{i=0}^{N-1}(1/\Delta f_i)^2} \tag{G.38}$$

then Equation G.37 becomes

$$\hat{t}_0 = \sum_{i=0}^{N-1} k_i t_0^i \pm \frac{\sigma}{\sqrt{\sum_{i=0}^{N-1}(1/\Delta f_i)^2}}. \tag{G.39}$$

From Equation G.37 it is seen that the LMS can be seen upon as estimating t_0 as linear combination of the measurements where the weights k_i is the optimum ones. It is also clear that the standard error of t_0 scales as

$$\sigma_{t_0} \propto \frac{\sigma}{\sqrt{N}}$$

Propagation of Errors for the Peak-Finder Since the PF requires the parameter estimates to be linear combination of the samples the standard error will also scale as $\sqrt{N}$. The weights α and β will not be the optimum ones, but for a large number of samples one will get results similar to the optimum parameter estimates given by the LMS.

G.3 Results

G.3.1 Efficiency

The differential time resolution was measured by illuminating two adjacent crystals with a LED. The light intensity was adjusted to give a light intensity of 400 ADC counts corresponding to $\sim$ 2 GeV. Date was taken with two different shaper options. The CERN Wuhan (CW) 4 μs option and the shaper developed by the Kurchatov institute (KU) with a rise time of 2 μs. (It was later discovered that the KU shaper was operated

with a wrong APD bias voltage so the results not be used to draw conclusion on the timing resolution). The data was analyzed bot with the Levenberg Marquardt nonlinear fit and the Peak-finder algorithm. Currently no good estimate of the error covariance matrix exists so the data was analyzed without taken into account an eventual correlation of errors.

As seen in Table G.3 the results obtained with a nonlinear fit and the PF algorithm is almost identical.

shaper	$\tau_{(1,14)}/\mu s$	$\tau_{(1,15)}/\mu s$	σ_{LM}/ns	σ_{PF}/ns
CW	4.248	4.267	2.147 ± 0.028	2.148 ± 0.028
KU	2.025	2.025	0.964 ± 0.015	0.948 ± 0.015

Table G.3: Results from LED runs in the PHOS lab taken in October 2004. The crystals+APDs were cooled down to -20 C^{circle} in a conventional freezer. 3000 events were analyzed. The Amplitude was $\sim$ 400 ADC counts

G.3.2 Simulation Results

The since the timing measurements given in Section **??** was differential it will not be possible to know if they are biased. A fit can give a biased estimate for Amplitude and TOF reconstruction if the model used for the fit does not match exactly the shaper response of the detection channel and/or if one makes wrong assumptions about the correlation matrix, such as for instance assuming that the error are equal and identical. For the PF algorithm there is an additional systematic error due to the fact that the overdetermined system of equations given by Equation6.9 and 6.5 cannot be solved exactly

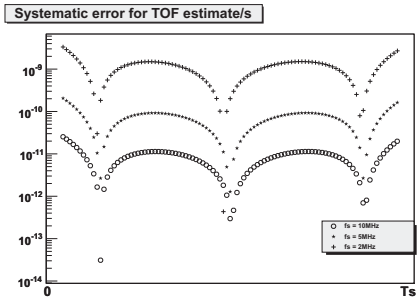
This section investigates some important properties of the PF algorithms. In Section G.3.2.1 it is investigated how the systematic error induced by PF algorithm depends on various parameters such as the sample rate, the rise time and the precision of the initial guess for the time delay. Knowledge about these properties are crucial when implementing an analysis chain.

G.3.2.1 Systematic Errors

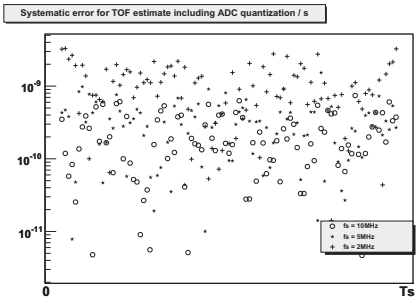
Dependence of Initial Guess of t0

G.3.2.2 Computing Speed

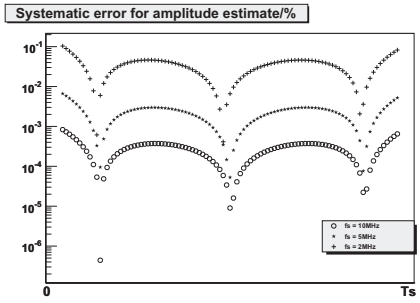
The efficiency of the PF algorithm was measured with simulated events. 500000 pulses G2 with tau = 2 us, amplitude 320 ADC counts and a time-delay of 15 ns with respect to the sampling clock was generated. The sampling rate was set to 2 MHz and pulses was sampled with 15 samples over the peak. The sampling rate of 2MHz is approximately



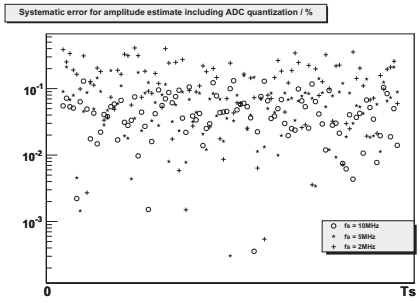
(a) The absolute value of the systematic error for TOF reconstruction for fs = 2, 5 and 10 MHz



(b) The absolute value of the systematic error for TOF reconstruction with fs = 2MHz, including ADC quantization noise

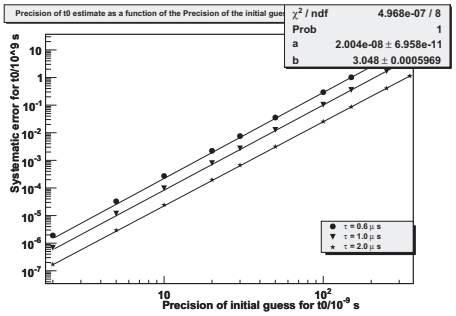


(c) The absolute value of the systematic error for amplitude reconstruction for fs = 2, 5 and 10 MHz

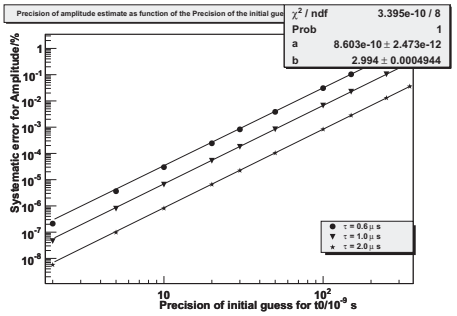


(d) The absolute value of the systematic error for amplitude reconstruction with fs = 2MHz, including ADC quantization noise

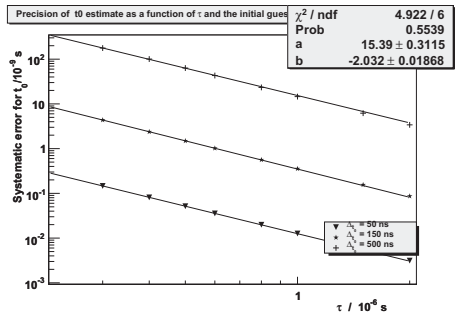
Figure G.6: The systematic error of the Peak Finder algorithm as a function of the time-of-flight.



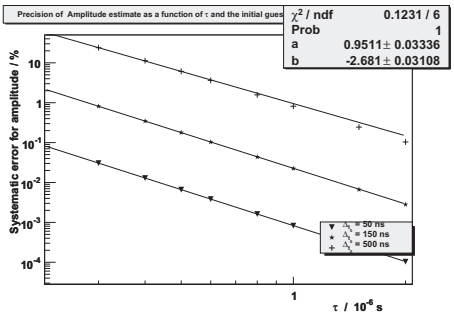
(a) The systematic error of TOF estimate as a function of the accuracy of the initial guess of t0



(b) The systematic error of the amplitude estimate by the as function of the accuracy of the initial guess of t0



(c) The systematic error for TOF reconstruction as function of the rise-time (tau) of the signal when the quality of the initial guess is held constant



(d) The systematic error for amplitude reconstruction as function of the rise-time (tau) of the signal when the quality of the initial guess is held constant

Figure G.7: The systematic error of the Peak Finder algorithm.

equal to the -3db cutoff frequency of the gamma-2 shaper so that the noise seen after ADC quantization can be assumed to be uncorrelated. The noise added in the simulation model was Gaussian white noise with an amplitude (one std) of 0.5 ADC counts. The results ares shown in Table G.4. As seen in Table G.4 the PF gives basically the same

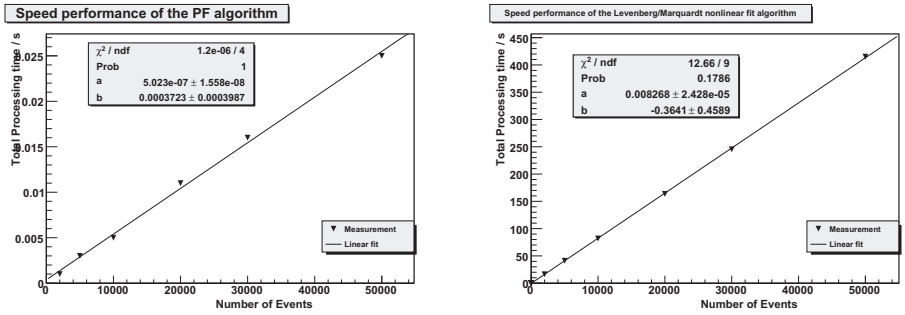
A_{inn}	$t0_{inn}/ns$	A_{LM}	A_{PF}	$t0_{LM}/nss$	$t0_{PF}/ns$
320	15	320.00 ± 0.26	320.01 ± 0.026	15.00 ± 1.39	14.91 ± 1.41

Table G.4: Results from simulations. 50000 events were generated and analyzed. The noise was set to 0.5 ADC count of Gaussian white noise. The amplitude A is given in entities of ADC counts.

results as the LM nonlinear fit. Therefore, if the LM fit is fully efficient, then the PF algorithm is also fully efficient.

G.3.2.3 Speed Performance Measurements

The processing speed per event of the LM and the PF was measured by measuring the time required to analyze various number of events. For each batch of events the total processing time was measured. Figure G.8(a) and G.8(a) shows the dependence of the total processing time to the number of events in each batch. The measurements are fitted with t = ax + b. The parameter b gives the overhead in starting up the analysis algorithm while a gives the processing time per event. Referring to Figure G.8(a) and G.8(b) the



(a) Processing speed performance of the PF algorithm

(b) Processing speed performance of the LM nonlinear fit

Figure G.8: Comparison of processing speed between the Peak Finder (PF) algorithm and the Levenberg Marquardt nonlinear fit (LM).

PF use about $500ns$ per event while the LM nonlinear fit spends $\approx 8ms$ per event. The

slope parameter (a) of the two graphs gives a ratio of processing speeds of

$$\frac{a_{LM}}{a_{PF}} = \frac{0.008268}{5.023 \cdot 10^{-7}} = 16460$$

Appendix H

Time-of-Flight Resolution Comparison with two Different Shaper Options

Table H shows results from time-of-flight measurements using the CW shaper developed at CERN/Wuhan and KU shaper developed at Kurchatov institute, Moscow. Measurements was performed using a nominal rise time of 2 and 4 μs for both shapers. The KU shaper was in addition tested using a rise time of 1.4 μs

As seen in the Table H the actual rise time measured on the physical signals are slightly higher than the nominal 2/4 μs.

The time-of-flight resolution was performed using differential timing between two channels denote ch#1 and ch#2, the numbering being arbitrary. Since time-of-flight resolution is most important at 2 GeV the LED and the pulser amplitudes was adjusted to give a signal $\sim$ 400 ADC counts so that the signal emulates a 2 GeV particle.

Differential time-of-flight measurement. The trigger for pulser and the LED outputs was not synchronized with the ADC sampling clock. The RCU always issues the trigger in sync with the ADC sampling clock and upon the arrival of a trigger that is not sync the RCU issues the trigger on the next rising clock edge. A direct measurement of the time-of-flight resolution would therefore give a uniform time-of-flight distribution in the range [0,..100] ns (assuming 10 MHz sampling) and its therefore not possible to measure the time-of-flight resolution directly for a single channel. *Differential time-of-flight measurement.* Two channels are injected with signal from the same source, for instance the pulser injecting the same signal to two channels the amplitude being similar in both channels. The time-of-flight difference is measured on an event by event basis giving a Gaussian distribution. In order to obtain the time-of-flight resolution for a single the width of the resulting distribution is divided by $\sqrt{2}$. It is assumed that the time-of-flight resolution for the two channels are the same and that the electronic noise is uncorrelated.

Fit Range. When analyzing data taken with different rise times the number of samples

used when fitting always covers the same range of the signal. If for instance a signal with $2\mu s$ rise-time is fitted using 80 samples then a signal with $4\mu s$ should be fitted using 160 samples in order to make a valid comparison of the results. An entry like 20 + 80 column *Fit range* means that the fit is performed using 80 samples starting from time bin 20. The start time bin is always chosen to be as close as possible to the real start of the signal. The start time bin is kept in the table just for book keeping.

Normalization. As seen in the table the amplitude varied quite significantly from the aimed 400 ADC counts. In order to facilitate the comparison of the results the rightmost column shows the results normalized to 400 ADC counts assuming a 1/E dependence ([43]) of the time-of-flight resolution on energy. A direct comparison between the CW and KU shaper is provided by run 3656, 3655, 3880 and 3885. As seen in the rightmost column the two shapers performs equally well for the 4 μs rise time option whereas the KU shaper is significantly better at 2 μs. By far the best result[1] is obtained with the KU shaper with a 1.4 μs rise time. It should be noted that the KU shaper is of first order whereas the CW is of second order. It should be pointed out that Table H does not give an accurate estimate of the time-of-flight resolution that can be expected for real signals (particle). For the pulser data charges are injected directly to the CSP bypassing the crystal and the APD so that the time-of-flight resolution is only affected by the electronic noise and the ADC digitization error. The time-of-flight resolution measured for pulser data is therefor better than what can be expected for real data. The rise time of the light output from the LED is much longer than real scintillation light from the crystals so that the time-of-flight resolution that can be obtained using LEDs is worse. The realistic time-of-flight resolution for real particle should therefore be somewhere in between the results obtained for LED and for pulser data.

Signal	Shaper	Run	$\tau/\mu s$		A/ADC cnt		Fit range	$\sigma_t/\sqrt{2}ns$	
			ch#1	ch#2	ch#1	ch#2			A=400
Pulser	CW	3623	2.38	2.36	455.8	457.8	20+80	$0.732^{\pm 0.016}$	$0.835^{\pm 0.018}$
Pulser	CW	3627	4.48	4.61	480.5	487.2	20+160	$0.737^{\pm 0.016}$	$0.891^{\pm 0.019}$
LED	CW	3656	2.2	2.2	431.0	401.9	20+160	$1.863^{\pm 0.042}$	$1.940^{\pm 0.044}$
LED	CW	3655	4.4	4.4	473.5	447.8	20+80	$2.516^{\pm 0.056}$	$2.897^{\pm 0.064}$
LED	KU	3878	1.44	1.43	470.7	467.6	15+50	$0.724^{\pm 0.016}$	$0.849^{\pm 0.019}$
LED	KU	3880	4.27	4.29	367.0	353.0	20+160	$3.191^{\pm 0.071}$	$2.872^{\pm 0.063}$
LED	KU	3885	2.20	2.18	362.3	348.2	17+80	$2.077^{\pm 0.046}$	$1.845^{\pm 0.041}$

Table H.1: Comparison between different shaper options.

[1] This was the original design

H.1 The Dependence of the Time-of-Flight Resolution on the Sampling Rate

The fact that PHOS signals are oversampled gives an interesting opportunity to demonstrate the meaning of *sufficient* statistics. The time-of-flight resolution was investigated using LEDs. The results is given in Appendix H.1. The aim her is not to evaluate these results but to present another measurement that was performed on run 3655. The measurement of the time-of-flight resolution was performed several times, each time discarding a data as follows; Firstly the number of samples used during the analysis was chosen in order to be divisible by as many integers as possible and at the same time enough to cover the whole signal. At 10 MHz 120 samples is a nice number. It is divisible by 1,2,3,4,5,6,8,10,12,15,20,24. This number of samples is used for data taken with the 4 μs shaper option. For data taken with the a shaping time of 2 μs or less either 60 or 64 samples was used. The analysis was done as follows: First all samples was used in the analysis. Then only every second sample, every third samples, every fourth sample and so on. This procedure emulates an reduced sampling rate. If for instance only every fourth sample is used then this is an emulation of a 2.5 MHz sampling rate.

Figure G.2 show the time-of-flight resolution as a function of the *fake* sampling frequency resulting from discarding data during the analysis. Referring to Figure 4.6 the cutoff frequency of the shaper is $\sim$ 1 MHz. The measurements indicated by the black squares are therefore fitted up to the Nyquist frequency of 2MHz. During the fitting it is assumed a $1/sqrt(f_s)$ dependence[43] of the time-of-flight resolution on the sampling frequency if the measurements are uncorrelated. The fit is indicated by the solid black line. The dotted blue line shows the extrapolation using the fit parameters from the [0-2] MHz range i.e the expected time-of-flight resolution if the measurements (samples) was always uncorrelated. As seen in the figure the time-of-flight resolution improves up to the $\sim$ Nyquist frequency but above 2 MHz there is almost no improvement. From the point of view of frequency analysis this can easily be explained by the fact it sufficient to sample at the Nyquist frequency in order to recover all frequency components of the signal. From the statistical point of view it can be explained as follows: The noise seen on the shaper output is always correlated and the correlation is determined by the cutoff frequency of the shaper. The correlation of the noise of the signal at two instances in time, lets say t_1 and t_2 is as a rule of thumb[66] correlated if the distance $t_2 - t_1$ is less than $1/f_{cutoff}$ where f_{cutoff} denotes the cutoff frequency of the shaper, i.e 1/2 the Nyquist frequency.

The question is what correlation one sees after digitization. The answer is that if one digitizes the signal with a sampling rate lower than the cutoff frequency one sees discrete random variables with correlated noise and if one samples higher than the cutoff frequency one sees correlated discrete variables. If now for instance, in the case of PHOS, sampling is done at 10 MHz, then increasing the sampling rate to 20 MHz would not improve the statistics since the additional data obtained at 20 MHz is correlated with

the data taken at 10 MHz. The new data could have been predicted from the already existing measurements since it is correlated with existing ones. In Table H.2 - H.5 the leftmost column shows the division factor D where for instance D=10 means that only every tenth sample was used. The second column show the emulated sampling frequency given by $f_s^{emulated} = 10MHz/D$. The last column shows the time-of-flight resolution obtained using the reduced sets of data.

D	$f_s^{emulated}/MHz$	Samples	σ_{TOF}
40	0.25	3	$7.866^{\pm 0.249}$
30	0.333	4	$4.871^{\pm 0.154}$
24	0.417	5	$3.781^{\pm 0.12}$
20	0.5	6	$3.298^{\pm 0.104}$
15	0.677	8	$4.923^{\pm 0.156}$
12	0.833	10	$2.592^{\pm 0.082}$
10	1	12	$2.611^{\pm 0.083}$
8	1.25	15	$2.486^{\pm 0.079}$
6	1.677	20	$2.381^{\pm 0.075}$
5	2	24	$2.745^{\pm 0.087}$
4	2.5	30	$2.401^{\pm 0.076}$
3	3.333	40	$2.440^{\pm 0.077}$
2	5	60	$2.332^{\pm 0.074}$
1	10	120	$2.315^{\pm 0.074}$

Table H.2: The dependence of time-of-flight resolution on the sampling rate for run 3655 (old numbering scheme) using 500 events and 120 samples for the analysis. Data was for the 4 μs CW shaper using LED with an amplitude of $\sim$ 400 ADC counts.

D	$f_s^{emulated}/MHz$	Samples	σ_{TOF}
15	0.667	4	$1.533^{\pm 0.034}$
12	0.833	5	$1.118^{\pm 0.025}$
5	2	12	$0.901^{\pm 0.02}$
4	2.5	15	$0.887^{\pm 0.02}$
3	3.333	20	$0.869^{\pm 0.156}$
1	10	60	$0.757^{\pm 0.017}$

Table H.3: The dependence of time-of-flight resolution on the sampling rate for run 3623 (old numbering scheme) using 1000 events and 60 samples for the analysis. Data was for the 2 μs CW shaper using pulser with an amplitude of $\sim$ 400 ADC counts.

D	$f_s^{emulated}/MHz$	Samples	σ_{TOF}
16	0.625	4	$1.357^{\pm 0.043}$
8	1.25	8	$1.940^{\pm 0.03}$
4	2.5	16	$0.827^{\pm 0.026}$
2	5	32	$0.755^{\pm 0.023}$
1	10	64	$0.733^{\pm 0.023}$

Table H.4: The dependence of time-of-flight resolution on the sampling rate for run 3623 (old numbering scheme) using 1000 events and 64 samples for the analysis. Data was for the 2 μs CW shaper using pulser with an amplitude of $\sim$ 400 ADC counts.

D	$f_s^{emulated}/MHz$	Samples	σ_{TOF}
16	0.625	4	$4.117^{\pm 0.092}$
8	1.25	8	$2.050^{\pm 0.046}$
4	2.5	16	$1.914^{\pm 0.043}$
2	5	32	$1.897^{\pm 0.042}$
1	10	64	$1.887^{\pm 0.042}$

Table H.5: The dependence of time-of-flight resolution on the sampling rate for run 3656 (old numbering scheme) using 1000 events and 64 samples for the analysis. Data was for the 2 μs CW shaper using LED with an amplitude of $\sim$ 400 ADC counts.

Appendix I

Results from FEE Transaction Tests

Table I.1 and I.2 shows the results from transaction tests for the board controller developed at Wuhan and the board controller developed in Bergen respectively. The test was performed by writing a value to a single register on either an ALTRO chip or on the board controller. If the value read back is not the same as the value written the transaction is said to have failed. The second last column shows failure rate given as the number of failed transtacction divded by the total number of transactions. The term transaction in this context is used to describe the one write/read of a rwegister. All possible mechanism of communication between the RCU and the ALTRO/board controller was tested: Board controller over the ALTRO bus (BCA), ALTRO over the ALTRO bus (AA) and board controller over the I^2C bus (BCSC).

Communication	Regadd.	Value	Branch	FEE No	Transaction	Failure rate	FEECs on
BCA	0x45	0x3ff	A	5	22200	0%	FEE 5
BCSC	0x45	0x3ff	A	5	12429	0%	FEE 5
AA	0x1	0x3ff	A	5	10400	0%	FEE 5
BCA	0x47	0x2aa	A	1	100	0%	All
BCA	0x47	0x2aa	A	2	100	0%	All
BCA	0x47	0x2aa	A	3	100	0%	All
BCA	0x47	0x2aa	A	4	100	0%	All
BCA	0x47	0x2aa	A	5	100	100%	All
BCA	0x47	0x2aa	A	6	100	100%	All
BCA	0x47	0x2aa	A	7	100	100%	All
BCA	0x47	0x2aa	A	8	100	100%	All
BCA	0x47	0x2aa	A	9	100	100%	All
BCA	0x47	0x2aa	A	10	100	100%	All
BCA	0x47	0x2aa	A	11	100	100%	All
BCA	0x47	0x2aa	A	12	100	100%	All
BCA	0x47	0x2aa	A	13	100	100%	All
BCA	0x47	0x2aa	A	14	100	100%	All
AA	0x1	0x2aa	A	1	100	0%	All
AA	0x1	0x2aa	A	2	100	0%	All
AA	0x1	0x2aa	A	3	100	0%	All
AA	0x1	0x2aa	A	4	100	0%	All
AA	0x1	0x2aa	A	5	100	100%	All
AA	0x1	0x2aa	A	6	100	100%	All
AA	0x1	0x2aa	A	7	100	0%	All
AA	0x1	0x2aa	A	8	100	100%	All
AA	0x1	0x2aa	A	9	100	0%	All
AA	0x1	0x2aa	A	10	100	0%	All
AA	0x1	0x2aa	A	11	100	0%	All
AA	0x1	0x2aa	A	12	100	0%	All
AA	0x1	0x2aa	A	13	100	0%	All
AA	0x1	0x2aa	A	14	100	0%	All
BCSC	0x47	0x2aa	A	1	256	0%	All
BCSC	0x47	0x2aa	A	2	256	0%	All
BCSC	0x47	0x2aa	A	3	256	0%	All
BCSC	0x47	0x2aa	A	4	256	0%	All
BCSC	0x47	0x2aa	A	5	256	0%	All
BCSC	0x47	0x2aa	A	6	256	0%	All
BCSC	0x47	0x2aa	A	7	256	0%	All
BCSC	0x47	0x2aa	A	8	256	0%	All
BCSC	0x47	0x2aa	A	9	256	0%	All
BCSC	0x47	0x2aa	A	10	256	0%	All
BCSC	0x47	0x2aa	A	11	256	0%	All
BCSC	0x47	0x2aa	A	12	256	0%	All
BCSC	0x47	0x2aa	A	13	256	0%	All
BCSC	0x47	0x2aa	A	14	256	0%	All
AA	0x1	0x2aa	A	3	7407	0.783%	All (bus replaced)

Table I.1: Results from transaction tests with the Wuhan board controller (version 0x30).

Communication	Regadd.	Value	Branch	FEE No	Transaction	Failure rate	FEECs on
BCA	0x65	0x2aa	A	1	260	1.923%	All
BCA	0x65	0x2aa	A	2	10	10%	All
BCA	0x65	0x2aa	A	3	10	50%	All
BCA	0x65	0x2aa	A	4	10	40%	All
BCA	0x65	0x2aa	A	5	10	50%	All
BCA	0x65	0x2aa	A	6	10	30%	All
BCA	0x65	0x2aa	A	7	10	10%	All
BCA	0x65	0x2aa	A	8	1000	0%	All
BCA	0x65	0x2aa	A	9	1000	0.400%	All
BCA	0x65	0x2aa	A	10	1000	0%	All
BCA	0x65	0x2aa	A	11	1000	0.400%	All
BCA	0x65	0x2aa	A	12	1000	0%	All
BCA	0x65	0x2aa	A	13	430	3.023%	All
BCA	0x65	0x2aa	A	14	10	30%	All
AA	0x1	0x2aa	A	1	1000	1.0%	All
AA	0x1	0x2aa	A	2	1000	0%	All
AA	0x1	0x2aa	A	3	1000	0.300%	All
AA	0x1	0x2aa	A	4	1000	0%	All
AA	0x1	0x2aa	A	5	1000	0%	All
AA	0x1	0x2aa	A	6	1000	0%	All
AA	0x1	0x2aa	A	7	1000	0%	All
AA	0x1	0x2aa	A	8	21	4.762%	All
AA	0x1	0x2aa	A	9	10	20%	All
AA	0x1	0x2aa	A	10	10	40%	All
AA	0x1	0x2aa	A	11	10	40%	All
AA	0x1	0x2aa	A	12	10	10%	All
AA	0x1	0x2aa	A	13	860	2.209%	All
AA	0x1	0x2aa	A	14	270	2.963%	All
BCSC	0x65	0x2aa	A	1	1024	0%	All
BCSC	0x65	0x2aa	A	2	1024	0%	All
BCSC	0x65	0x2aa	A	3	1024	0%	All
BCSC	0x65	0x2aa	A	4	1024	0%	All
BCSC	0x65	0x2aa	A	5	1024	0.098%	All
BCSC	0x65	0x2aa	A	6	1024	0%	All
BCSC	0x65	0x2aa	A	7	1024	0.098%	All
BCSC	0x65	0x2aa	A	8	1024	0%	All
BCSC	0x65	0x2aa	A	9	1024	0%	All
BCSC	0x65	0x2aa	A	10	1024	0%	All
BCSC	0x65	0x2aa	A	11	1024	0.098%	All
BCSC	0x65	0x2aa	A	12	1024	0%	All
BCSC	0x65	0x2aa	A	13	1024	0%	All
BCSC	0x65	0x2aa	A	14	1024	0%	All

Table I.2: Results from transaction tests with the Bergen board controller (version 0x34).

Appendix J

Local Guide to the PHOS/EMCAL APD Bias Control (PABC) software

Local Guide to the PHOS/EMCAL APD Bias Control (PABC) software

Version 4.1

Per Thomas Hille

Department of Physics, University of Oslo
2 October 2006

Abbreviations

```
APD           = Avalanche Photo Diode
CSP           = Charge Sensitive Pre-amplifier
DCS           = Detector Control System
DCS card      = Detector Control System card
DID           = DIM Information Display
DIM           = Distributed Information Managment
DIM_DNS       = DIM Dynamic Name Server
FEE           = Front End Electronics
FEEC          = Front End Electronics Card
GUI           = Graphical User Interface
HV            = High Voltage
LV            = Low Voltage
PABC          = Phos APD Bias Control
PC            = Personal Computer
PHOS          = Photon Spectrometer
RCU           = Readout Control Unit
```

J.1 Introduction

This is the User Manual for the PHOS APD Bias Control (PABC) Front End Electronics FEE configuration software. The PABC GUI interacts graphically directly with the software libraries that will be used for ALICE/PHOS DCS in the LHC. The communication with the PHOS detector is based on DIM.

The PHOS DCS software is separated in two parts. The Configuration part and the monitoring part. This manual covers the FEE configuration part.

The PHOS DCS FEE configuration software package consists of standalone C++ classes to configure the PHOS FEE and a graphical C++ layer to interact graphically with these classes. The graphical software layer and the configuration software layer is completely separated so that the FEE configuration software can be interfaced either from the PABC GUI, from PVSS, from the command line, or from any other program.

The PABC GUI is intended for commissioning and calibration of the PHOS detector. For the final system the interaction with the PHOS DCS software will be via PVSS. The PABC GUI however is made in ROOT. The reason for the decision of making this GUI in ROOT and not in PVSS was that the requirements for commissioning and calibration is different from the requirements in the final system. More specific one needs much more detailed control of the configuration during calibration and commissioning than during physics data taking.

J.1.1 Terms, Items and definitions

J.1.1.1 The Readout Control Unit (RCU)

The RCU is responsible for reading out the front end electronics. The RCUs is situated just outside the PHOS module and is connected to 28 front end cards situate inside the module. A total of four RCUs is necessary in order to read out one PHOS module. Before reading out data the RCU must know which channels to read out, the front end cards must know how many samples to take for each event/trigger. In addition the PHOS Monitoring and Safety Module (PCM) of each of the cards must know which APD bias DAC settings to apply. The configuration of the FEECs is the responsibility of the RCU. The RCU receives the FEE configuration from the PABC GUI via a DIM server (fee-server) running on the DCS card and it is the responsibility of the RCU to apply these settings to the electronics.

J.1.1.2 DCS card

The DCS card is a mezzanine card that is connected directly to the RCU. The task of the DCS card is to configure the RCU, to tell the RCU which trigger input to use etc and to give the RCU the configuration of the FEE. The DCS card includes a FPGA with an embedded arm processor. The arm processor on the FPGA runs under a Linux operating system (Busy Box). This makes it possible to communicate with the DCS card via Ethernet. From the point of view of the network the DCS card looks like any other PC.

J.1.1.3 Distributed Information Management DIM

Most of the information given about DIM in this section is taken unmodified from [67]. The PHOS DCS software is based on DIM. DIM was developed at CERN for the Delphi experiment and is widely used in several other experiments at CERN. DIM, like most communication systems, is based on the client/server paradigm.

The basic concept in the DIM approach is the concept of "service". Servers provide services to clients. A service is normally a set of data (of any type or size) and it is recognized by a name - "named services". The name space for services is free. Services are normally requested by the client only once (at start-up) and they are subsequently automatically updated by the server either at regular time intervals or whenever the conditions change (according to the type of service requested by the client). The client updating mechanism can be of two types, either by executing a callback routine or by updating a client buffer with the new set of data, or both. In fact this last type works as if the clients maintain a copy of the server's data in cache, the cache coherence being assured by the server.

J.1.1.4 The DIM DNS server

In order to allow for transparency. as well as to allow for easy recovery from crashes and migration of servers, a name server is used. Servers "publish" their services by registering them with the name server (normally once, at start-up). Clients "subscribe" to services by asking the name server which server provides the service and then contacting the server directly, providing the type of service and the type of update as parameters. The name server keeps an up-to-date directory of all the servers and services available in the system. Figure J.1 shows how DIM components (Servers, Clients and the Name Server) interact. Whenever one of the processes (a server or even the name server) in the system crashes or dies all processes connected to it will be notified and will reconnect as soon as it comes back to life. This feature not only allows for an easy recovery, it also allows for the easy migration of a server from one machine to another (by stopping it in the first machine and starting it in the second one), and so for the possibility of balancing the machine load of the different workstations.

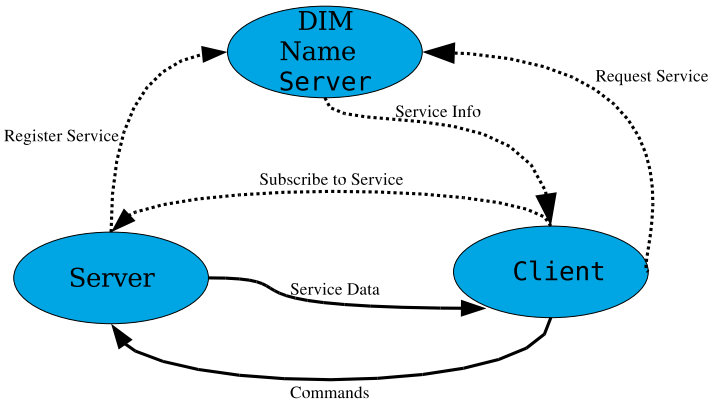


Figure J.1: Communication via DIM

J.1.1.5 The Fee-server

The fee-server consist of a DIM server and a software package specific to PHOS. The PHOS specific software is called the Control Engine (CE) and contains several services

specific to PHOS. One such service is for instance to run scripts on the DCS card.

The DIM server and the CE combined is called a fee-server. The fee-server runs on the DCS cards and takes commands from the fee-client (the PABC GUI) running on a PC, and sends the results of various commands back to the PABC GUI.

J.1.1.6 The Control Engine (CE)

The control engine (CE) runs on the DCS cards an is a part of the fee-server software. The CE interprets the content of data sent via DIM and also provides service data to the fee-client (for instance temperatures of the electronics). In addition the CE takes care of the communication between the RCU and the front end electronics.

J.1.1.7 The Fee-client

The fee-client is a DIM client with some additional functionality. There is also a PHOS version of this client. In the DCS software this client is called the PhosFeeClient. From now on we will refer to this client as the fee-client.

J.1.2 The PHOS DCS infrastructure

Figure J.2 shows schematically the communication infrastructure for a single PHOS module. When the fee-server on one of the four DCS cards starts it automatically register itself and its services on the DIM DNS server. When the PABC GUI starts up it automatically starts a fee-client that communicates with these fee-servers. There is only one such client in the system. The client requests services from the fee-servers running on the DCS cards by contacting the DIM DNS node. The DIM DNS node replies by giving the service information to the client. Thereafter the client and the servers communicates directly, independently of the DIM DNS node.

There is four readout partitions on a single module. Correspondingly it is indicated a one to four relationship between the fee-client and the fee-server. The Figure would have looked almost the same if there were more than one, lets say N PHOS modules except that there would have been a one to Nx4 relationship between the fee-client and the fee-server. No matter the number of PHOS modules there will never be more than one[1] fee-client to the fee-servers.

Once the communication chain is established the user can send commands to the RCU and the FEE via the fee-servers. The fee-client can also receive service data. Such service data might for instance be temperatures, voltages, and currents of the front end cards.

[1] Unless you start the PABC GUI from more than one terminal

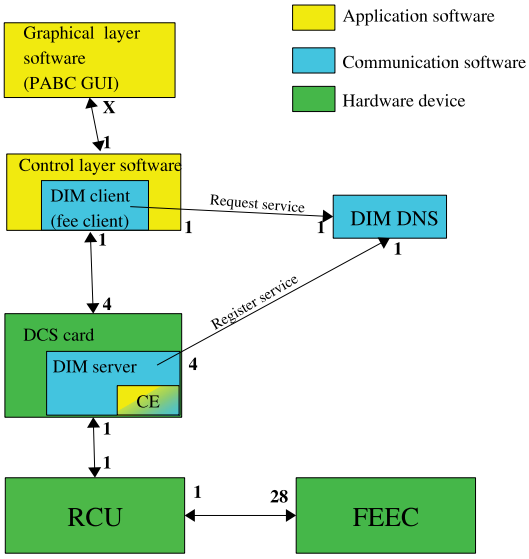


Figure J.2: The PHOS DCS Communication infrastructure via DIM for a single PHOS module.

J.2 Setting up the APD bias control from scratch

This chapter explains how to set up the communication chain for the PHOS APD bias control from scratch. If the communication chain already works the procedures in this chapter are not necessary. To set up the communication chain requires several steps listed below. The steps must be done in the given order.

1. Ramp down HV for the front end electronics.
2. Ramp down LV for the front end electronics.
3. Turn off the voltage for the RCU/DCS.
4. Turn on the voltage for the RCU/DCS.
5. Reboot the PC that will run the DIM_DNS server + the PABC +DID
6. Make sure that the DIM DNS node is running.

7. Start the DID information display (optional)
8. Make sure that a fee-server is running on each of the DCS cards.
9. Start a fee-client that communicates with the fee-servers on the (This will be the PABC GUI).
10. Ramp up LV for the front end electronics.
11. Make sure the LV for the CSP preamplifiers are ON[2].
12. Apply APD configuration.
13. Ramp up HV for the front end electronics.

If you want to apply new APD settings to the electronics it must be done **before** ramping up the HV for the APDs. Otherwise the HV power-supply can trip

Steps that are governed by the PABC software will be explained below. Turning on/off voltages etc will not be explained. There are in addition several other steps necessary to get a full communication chain working. These steps are done automatically provided that you use the following PC and DCS cards.

- The DIM DNS node, and the DID information display runs on the PC *alphspcdcs01.cern.ch* under user **phos**
- The DIM servers runs on the DCS cards, all as user root. Which DCS cards to use is subjected to changes. The currently mounted DCS/RCUs are listed below.

  alphsdcs0060.cern.ch #DIM name **alphsdcs0060**
  alphsdcs0279.cern.ch #DIM name **alphsdcs0279**
  alphsdcs0281.cern.ch #DIM name **alphsdcs0281**
  alphsdcs0282.cern.ch #DIM name **alphsdcs0282**

- The PABC GUI runs on the PC

  alphspcdcs01.cern.ch

If you want to use different DCS cards than the ones listed above please contact the author at *perthi@fys.uio.no*.

[2]The reason is that the voltage for the DACs holding the APD bias settings is derived from the 13.7 Voltage for the CSPs

J.2.1 Preparing to start from scratch

Reboot alphspcdcs01 In order to make sure that there are no background processes that can cause trouble, if you are not already logged in then log in either via the network or directly to the PC.

- Turn off the power for the DCS cards
- Reboot *alphspcdcs01*

  From the network

  ```
  [any machine]  > ssh phos@alphspcdcs01.cern.ch

  [any machine]  > username: phos

   phos@alphspcdcs01's password: > password: phosbeamtest

  [alphspcdcs01] /home/phos > reboot
  ```

  Directly on the PC

  ```
  user: phos

  username: phosbeamtest

  [alphspcdcs01] /home/phos > reboot
  ```

  Please note that you will lose the connection to *alphspcdcs01* when you reboot.
- Reboot the DCS cards *alphsdcs0060, 0279, 0281, 0282* by turning ON the power.

J.2.2 Starting the DIM name server

The DIM DNS node should start up automatically when rebooting *alphspcdcs01.cern.ch*. If you really want to make sure that the DNS node is running go to Chapter J.5.1.

J.2.3 Starting the DIM Information Display (optional)

The DIM Information Display (DID)is not mandatory but a very useful tool for debugging and monitoring of the DIM communication. It can run on *alphspcdcs01* or any other machine that has the necessary software and environment. Note that on some occasions the DID will crash when run remotely due to a bug in DID. If this happens you can run it directly on *alphspcdcs01*. To start DID do the following.

```
[any machine ~] >  ssh -X phos@alphspcdcs01.cern.ch

 phos@alphspcdcs01's password: phosbeamtest

[alphspcdcs01] /home/phos > . dcs_cern.sh        ##sets up the enviroment

[alphspcdcs01] /home/phos > did
```

Now the DID information display will show up. To view all available servers select "View" from the menu and then "all servers". It is useful to also select from the menu "Commands" and then "Set Debug On". Currently the display will only show one server, the DIM DNS name server.

J.2.4 Starting the DIM servers on the DCS cards

If the DCS cards are mounted on the dedicated private network on the PHOS DCS gateway (*alphspcdcs01.cern.ch*) then the fee-servers should start automatically when the DCS cards are powered on. If the DCS cards are accessible on the CERN network then the fee-servers must be started manually. In the latter case the procedure described in this section must be repeated for the four DCS cards *alphsdcs0060, 0279, 0281, 0282*.

The procedure will be shown only for *alphsdcs0279.cern.ch*. The procedure is identical for *alphsdcs0060.cern.ch, 0279, 0281, 0282*.

To start the DIM servers do the following.

```
[any machine ~] > ssh root@alphsdcs0279.cern.ch

root@alphsdcs0279's password: dcs

alphsdcs0279:\$ ./home/phos/feeserver_start.sh
```

After all the four DIM servers are started you should see five servers in the DID information display. One server is the DIM DNS node, and the other four are the servers running on the DCS cards. The DID display should look as shown in Figure J.3.

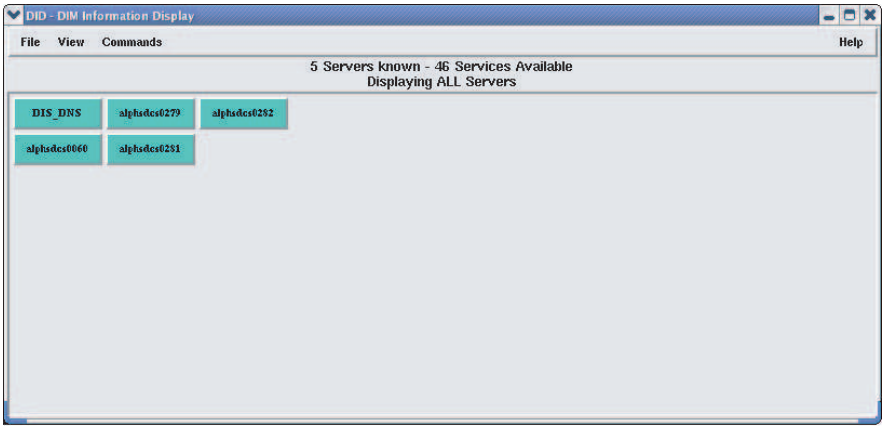


Figure J.3: The DIM Information Display (DID).

J.2.5 Starting the APD GUI (The DIM client)

To start the PABC bias control GUI do the following.

```
[any machine ~] >  ssh phos@alphspcdcs01.cern.ch

phos@alphspcdcs01's password: phosbeamtest

[alphspcdcs01] /home/phos > . dcs_cern.sh        ##sets up the enviroment

[alphspcdcs01] /home/phos > cd phos_dcs

[alphspcdcs01] /home/phos > apdgui
```

J.3 Using the PABC Graphical User Interface (apdgui)

This chapter gives an explanation on how to use the various options of the APD bias control GUI. It is assumed that the communication chain is already set up and working.

J.3.1 Starting the PABC Graphical User interface

Log in to *alphspcdcs01.cern.ch* with user **phos** and password **phosbeamtest**. Make sure that the fee-servers and the DIM DNS server are running (see Chapter J.2.2 and J.4.9 for details). Before starting the PABC GUI you must set up the environment variables.

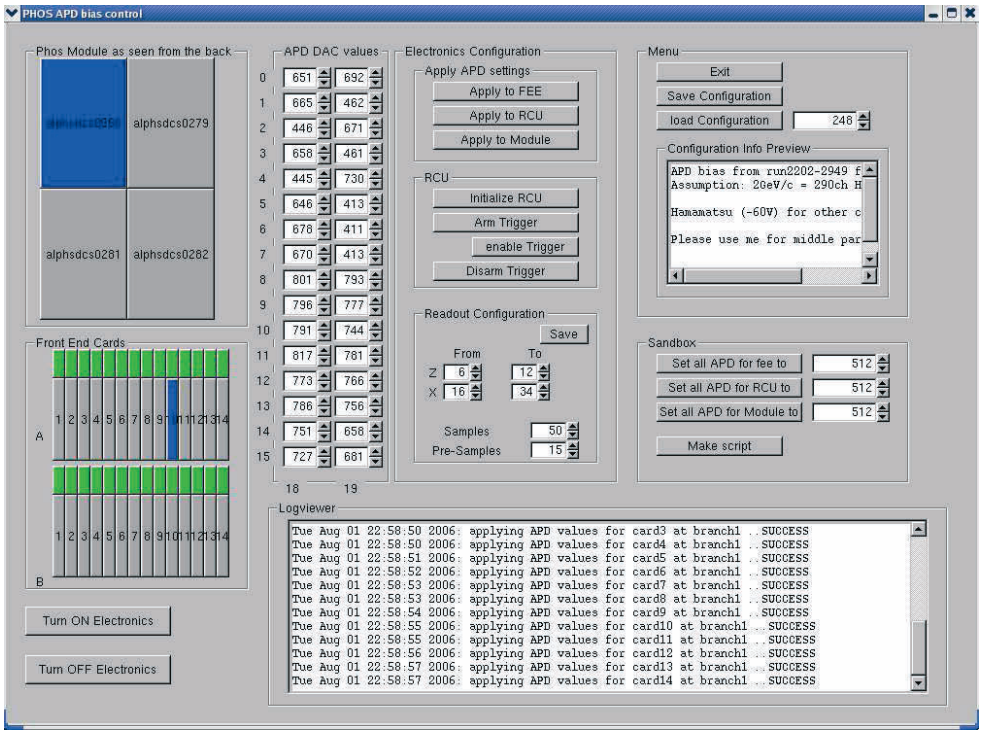


Figure J.4: The PABC Graphical User Interface (GUI)

```
[alphspcdcs01] /home/phos > . dcs_cern.sh
```

To start the GUI do

```
[alphspcdcs01] /home/phos > cd phos_dcs/
[alphspcdcs01] /home/phos/phos_dcs > apdgui
```

J.3.2 Actions of the PABC program at start-up

When the PABC is started (see section J.2.5) it performs the following actions.

- It initializes a fee-client.
- It to establish contact with DIM DNS server and asks for the services of the fee-servers running on the DCS cards, it will use the DIM DNS server defined in the setup script **dcs_cern.sh** the folder */home/phos*.
- It attempts to establish contact with the fee-servers.
- It loads the latest configuration of the APD bias settings from the folder */home/phos/data/ConfigurationFiles*
- It puts a copy of the latest configuration in to the folder */home/phos/data/APD_settings*.
- It loads the values from */home/phos/data/APD_settings* into the GUI to display it to the user.

J.3.3 The PABC GUI User menus and their actions

The PABC user panel has a number of different buttons to manipulate the APD configuration and apply it to the FEECs. In addition it is possible to reset the RCU and to configure which crystals to read out during data taking.

The user actions are separated into several Menu groups. Different menu groups have different domains of actions. The available menu groups and their domain of actions is explained below. A detailed description is given in the following sub chapters.

- **Menu:** This is the main menu that deals with the interaction with the database. It loads configurations from the database into the local working directory and saves configuration from the local directory to the database.
- **Sandbox:** The word *sandbox* refers to your local playground. Actions in this menu group only applies to files in the local working directory */home/phos/data/APD_settings*.
- **Electronics Configuration:** This the menu that interacts directly with the electronics. It has several sub-menus.

  Apply APD settings: Applies the settings in the local working directory to the electronics. Please note that applying the settings to the electronics does not store the data in the database. If you want to save the settings that was applied, use the save configuration button in the Main menu.

  RCU: Menu to reset RCUs and to configure trigger settings.

  Readout Configuration: Menu to select readout region.

J.3.3.1 Menu: PHOS Module as seen from the back

The PHOS module is divided into four readout partitions. The name of the partitions is identical to the DIM name of the fee-server used to communicate with that partition. At the start-up of the GUI the color of all the RCU partitions will be grey which means that no RCU is selected. The user selects one of the RCUs by a single click with the mouse. Clicking one of the RCU partition buttons of the PHOS Module will perform the following actions:

- The PABC will contact the fee-server (DIM server) controlling that readout partition and ask for the active front end card list of selected RCU.
- The PABC will try to contact all the cards marked as active and if contact is established, it sets the state of the corresponding FEEC to ON. If contact is not established the state is set to unknown (grey).
- The color of the selected RCU will be set to blue.

After all the FEECs in the active front-end card list of the selected RCU is accessed the GUI will draw all the FEECs of the currently selected RCU on the bottom left of the GUI. Please note that it will take several seconds for the PABC to probe all cards.

J.3.3.2 Menu: Front End Cards

One front-end card consists of two parts. One small button to turn on or off the card, and one long button to select the card in order to configure the APD settings. The "Front end card" menu shows all the FEECs for the currently selected RCU.

The ON/OFF button By clicking on the ON/OFF button the user can turn the card ON or OFF. The color of this button indicates the state of the card. A click on the button will turn the card OFF if it is already ON, otherwise it will be turned ON provided that the PABC is able to contact the card. The meaning of the various colors of the ON/OFF button is explained below.

- *Grey*: The state is unknown. It means that The RCU did not manage to contact the card or that the RCU has been reset. If the RCU has just been reset (from the GUI) then all cards will be grey.
- *White*: The card is present, but power is OFF.
- *Green*: The card is ON and ready to take commands from DCS.
- *Red*: The card is in the state ERROR.

Please note that the status of the cards are only updated on selection of an RCU or when the ON/OFF buttons are clicked. If one has interacted with the corresponding DCS card manually by for instance running scripts on the command line of the DCS card, then the state of the card might not correspond to the state shown in the PABC GUI. To get the correct display of the FEE states in this case will require the PABC GUI to be closed and then started again.

The select FEEC button The long button below each ON/OFF button is the button to select the corresponding card for APD configuration. Clicking on a FEE button will perform the following actions:

- The current APD settings will be read from folder */home/phos/data/APD_settings*. If no file exists, a file with default values will be created. The current default value is DAC=512 for all APDs which corresponds to ≈300 Volt on the HV output.
- The color of the card will be set to blue.
- A 2x16 matrix of APD DAC values will be drawn in the middle of the GUI.
- If a card was already selected and the user selects another card, then the APD settings of the previous card will be saved to the corresponding file in the folder */home/phos/data/APD_settings*

J.3.3.3 Menu: APD DAC values

In this menu the user can change the DAC values of the individual APDs. This is done either by clicking the up down arrows to the right of the number entry, or by typing a number in the number entry field and pressing *enter* on the PC keyboard.

By using the up down arrows it will only be possible to set value's in the range 0 - 1023. If one types a number outside this range in the number entry field then: if the number is higher than 1023 then it will be set to 1023 and if the number is less than zero it will be set to zero.

J.3.3.4 Menu: Menu

- **Exit:** Not yet implemented
- **Save Configuration:** Saves everything in the folder */home/phos/data/APD_settings* in a single file in the Folder */home/phos/data/ConfigurationFiles*. Please note that clicking this button will never overwrite any data. The ID number , N, of the latest configuration will be read from the file */home/phos/data/ConfigurationFiles/Id.txt* and the new configuration file will be written to the file */home/phos/data/ConfigurationFiles/configuration_N+1.txt*.

- **Load Configuration:** Loads the configuration with the ID given by the user specified number to the right of the button. The default value of the number entry is the configuration that was loaded at start-up of the *apdgui*. If the given configuration does not exist an error is issued on the command line,and the data in */home/phos/data/APD_settings* is left untouched.

J.3.3.5 Menu: Sandbox

- **Set all APD for FEE to:** Clicking this button will set all the APD values for the currently selected FEE to the value given in the entry to the right of the button. The values will be written to the sandbox /home/phos/data/APD_settings. Please note that the changes will not be updated in the database unless you click the **Save Configuration** button.
- **Set all APD for RCU to:** Clicking this button will set all the APD values for the currently selected RCU to the value given in the entry to the right of the button. The values will be written to the sandbox */home/phos/data/APD_settings*. Please note that the changes will not be updated in the database unless you click the **Save Configuration** button.
- **Set all APD for Module to:** Clicking this button will set all the APD values for the currently selected Module (Module 2) to the value given in the entry to the right of the button. The values will be written to the sandbox */home/phos/data/APD_settings*. Please note that the changes will not be updated in the database unless you click the **Save Configuration** button.

J.3.3.6 Menu: Electronics Configuration

This menu groups contains several sub menus. Actions in this menu group is applied to the electronics.

Warning: Make sure that High Voltage is turned off before making any changes to the APD/DAC values. The High Voltage must be ramped up **after** the APD settings is applied.

Apply APD settings

- **Apply to FEE:** Applies the setting shown in the 2x16 matrix of crystals/APDs that is currently selected. The currently selected FEEC is marked with blue (by click from the user). If no card is selected an error is issued on the command line. Please note that clicking this button only attempts to apply the settings for the selected FEEC as it is given in the folder */home/phos/APD_settings*, it does not store any data in the configuration database.

- **Apply to RCU:** By clicking this button all the APD settings for the currently selected RCU in the "sandbox" directory (/home/phos/data/APD_settings) is applied to the FEE. The settings is applied one by one on FEEC basis.
- **Apply to module:** Disabled for the moment.

RCU All buttons in this menu group perform actions on all RCUs of the PHOS module except the **Initialize RCU** button that initializes only the currently selected RCU (the one colored blue).

- **Initialize RCU:** Performs the following actions:

  Resets the the RCU driver.

  Resets the DCS firmware.

  Reloads the firmware for the currently selected RCU. (equivalent to the command 0xb000 0x4).

  Resets the RCU firmware.

  Gives the DCS card access to the readout bus (equivalent to the command 0xe000 0x0)

- **Reset All RCUs:** Performs the following actions.

  Reloads the firmware for all RCUs (equivalent to the command 0xb000 0x4)

  Resets the RCUs

  Powers down all FEECs

  Gives the DCS card access to the readout bus (equivalent to the command 0xe000 0x0). WANING: all the FEECs will be turned off and any configuration such as for instance the APD settings of the cards will be lost. After clicking the **initialize RCU** button. The card will have to be turned on and the APD bias settings must be applied again.

- **Arm Trigger** performs the following actions.

  Reads the number of samples + pre-trigger samples from the GUI an applies it to the electronics.

  Calculate a suitable trigger window between L1 and L2 trigger from the number of samples and applies it to the electronics.

  Generates the active channel lists (AFL) for the four readout partitions to read out based on the readout region specified by the user.

  Distribute the AFLs to the four readout partitions (RCUs) and applies them to the electronics.

- **Enable Trigger** Enables triggers for all four readout regions.
- **Disarm trigger** Disables the external hardware trigger and resets the RCU.

  Tells the RCU that it is a PHOS type RCU.

  WARNING: Clicking the **Disarm Trigger** button resets the RCU and all registers in the RCU including the register that specifies if the RCU is TPC type or PHOS type. The mapping of the power on lines of the GTL bus is different for TPC and PHOS. After a reset the register telling which type of RCU (register 80009) is set to the default value which is TPC. Unless ALL FEEC is powered ON when resetting the RCU the APD settings will be gone and the FEEC APD bias settings must be reconfigured. If all FEECs is turned on when disarming the trigger then it does not matter if the mapping is changed during reset and the settings of the FEECs is not lost.

Readout Configuration In the readout configuration menu the user specifies the number of samples, the number of pre-trigger samples to use. The readout region specifies the matrix of crystals to read out. The Z axis is horizontal and the X axis is vertical. Origo (0,0) is in the upper left corner and the coordinate (55, 63) is in the bottom right corner of the PHOS module as seen from the back (looking at the electronics). Changes made in the "Readout configuration" menu takes effect when the user clicks the **Arm Trigger** button. If you want to retrieve the readout configuration automatically the next time you open the PABC GUI then click the **save** button.

J.3.3.7 The LogViewer

The LogViewer display various messages from the system as response to user actions. The messages shown in the LogViewer is also written to the file */home/phos/phos_dcs/log.txt*.

J.4 PABC How To...

J.4.1 Turn ON/OFF a FEEC

To turn ON/OFF a FEEC simply click the ON/OFF button for the card. The ON/OFF button of the FEEC is the small button on top of each of the cards. Clicking the ON/OFF button will attempt to turn the FEEC OFF if it is already on, otherwise the PABC will attempt to turn it ON. If you successfully turned ON the FEEC the ON/OFF button will turn green as indicated in Figure J.4. The LogViewer will issue the message.

```
CardX at branchY was turned ON, pcmversion is 0x20
```

If you get any other messages[3] you can try to do it once more. If you still don't get this message go to section J.6.2

J.4.2 Apply APD settings to a single FEEC

To apply APD settings to a FEEC the card must be turned on (see Section J.4.1), and the card you want to apply the APD settings to must be selected. To select the card just click on the card. The card should turn blue and the current APD settings of the card will be shown in the 2x16 matrix to the right. The PABC GUI should look something like Figure J.4 Please note that the entries in the 2x16 matrix does not necessarily reflects the APD bias settings of the actual front end card unless you apply the settings successfully to the electronics.

To apply the settings of the currently selected FEEC in the GUI to the electronics click the **Apply to FEE** button in the **Apply APD settings** sub-menu under the **Electronics Configuration** menu. If the APD settings was applied correctly the LogViewer will issue the message

```
applying APD values for cardX at branchY ..SUCCESS
```

If you get any other message go to Chapter J.6.7

J.4.3 Apply APD settings to all cards on readout partition

The readout partition (RCU) you want to apply the APD bias settings too must be selected. An readout partition is selected if it has the color blue. To apply the current APD configuration click the button **Apply to RCU** button.

J.4.4 Apply APD settings to all cards of the Module

Click the **Apply to Module** button in PABC GUI.

J.4.5 Take data

Before attempting to take any data please make sure that ALL FEECs of all four readout partition is ON even if you are only taking data from a part of the PHOS module. There is several issues/conflicts between DAQ and DCS. The problem is that DAQ occasional leaves the RCU in a state of error when it is impossible to access it from DCS or to take any data. There is however a recipe to follow that will keep the system working both for DCS and DAQ/DATE. The recipe is given below.

[3]The PCM version number is subjected to changes

Recipe for taking data

1. PABC: click the **Disarm Trigger** button if it was not already done after previous run. If you are not sure if you clicked the button just click it in any case.

2. PABC:(optional) apply new APD bias settings.

3. PABC: click the **Arm Trigger** button

4. DATE: click **Start Process** and the **Start**.

5. PABC: click the **Enable Trigger** button.

6. Wait until you have enough events from DATE.

7. PABC: click the **Disarm Trigger** button.

8. DATE: click **Stop**.

J.4.6 Initialize the RCU(s)

WARNING: before clicking the **initialize RCU** button please make sure that the LV voltage for the FEE and CSPs and the HV is OFF. The RCU is initialized every time the DCS cards is rebooted or the power for the RCU is switched on. It should therefor normally not be necessary manually to initialize the RCU. However it might be that for various reasons the RCU must be re-initialized.

To initialize an RCU first click on the readout partition corresponding to the RCU that you want to initialize. After selecting the RCU, wait a few seconds to let the DCS card probe the electronics. Then click the **"Initialize RCU"** button in the RCU menu group in the PABC GUI. This will initialize the currently selected RCU (the one marked with blue). If no RCU is selected you will get an error message from the Logviewer. If you want to initialize all four RCUs you will have to select them one by one and repeat the procedure.

J.4.7 Log in to a DCS card

The fee-servers can run either on the CERN network or on a dedicated private network on the PHOS DCS gateway (currently alphspcdcs01). The default is that the fee-servers are running on the gateway, however for debugging purposes they might be run on the CERN network.

J.4.7.1 Log in to a DCS card running on the CERN network

To log in to a DCS card do the following:

```
[any PC ~] ssh root@alphsdcsXXXX
 root@alphsdcsXXXX's password: dcs
```

Here XXXX represents the serial number of the DCS card. For the moment the available DCS cards is *alphsdcs0060, 0279, 0280, 0281, 0282*. For instance, if you want to log in to the DCS card with serial number 0279 replace XXXX with 0279 and do the following.

```
[any machine ~] > ssh root@alphsdcs0279 ssh

root@alphsdcs0279's password: dcs
```

J.4.7.2 Log in to a DCS card running on the protected network on the DCS Gateway

If the DCS cards are running on the dedicated network on the DCS gateway then the DCS cards will not be visible on the CERN network. It is necessary to first log in to the gateway, and then log in to the DCS cards from there. The DCS gateway currently used is *alphspcdcs01*. For instance to log in to the *alphsdcs0279* do the following:

```
[any machine ~] > ssh phos@alphspcdcs01.cern.ch

phos@alphspcdcs01's password: phosbeamtest
[alphspcdcs01] /home/phos > ssh root@alphsdcs0279

root@alphsdcs0279's password: dcs
```

J.4.8 Check if the fee-server is running

First log in tho the DCS card where you want to check if the fee-server is running (see Section J.4.7). On the command line of the DCS card type the command **ps -a**.

```
alphsdcsXXXX:/ \$   ps -a
```

This will write out the complete list of processes running on the DCS card (the **-a** means *all*). The list will look something like the one below.

J.4.8.1 If the fee-server was started manually

If the fee-server is running there will be three processes associated with it. You should have under the column "Command" three entries with the name **/bin/feeserver**. If you started the feeserver with the start-up script (the normal way) there will also be an entry called **/bin/sh ./home/phos/feeserver_start.sh**. The list of processes will look something like the one below.

```
  PID  Uid     VmSize Stat Command
    1 root        356 S   init
    2 root            SW  [keventd]
    3 root            SWN [ksoftirqd_CPU0]
    4 root            SW  [kswapd]
    5 root            SW  [bdflush]
    6 root            SW  [kupdated]
    7 root            SW  [mtdblockd]
    8 root            SWN [jffs2_gcd_mtd3]
   34 root        340 S   /sbin/syslogd -m 0 -O /var/log/messages
   35 root        296 S   /sbin/klogd
   38 root        396 S   /sbin/devfsd /dev
   65 root        360 S   udhcpc -b
   70 root        908 S   /usr/local/sbin/sshd -D
   75 root        352 S   /usr/sbin/inetd
   76 root        360 S   /usr/local/sbin/boa
   80 root        348 S   /usr/sbin/crond -l 9 -c /etc/crontab/
   86 root        216 S   /usr/local/sbin/writer
  113 root        416 S   -sh
  116 root       1180 S   /usr/local/sbin/sshd: root@pts/0
  125 root        428 S   -sh
  971 root        372 S   /bin/sh ./home/phos/feeserver_start.sh
  972 root        384 S   /bin/feeserver
  973 root        384 S   /bin/feeserver
  974 root        384 S   /bin/feeserver
  982 root       1160 S   /usr/local/sbin/sshd: root@pts/1
  984 root        416 S   -sh
 1010 root        340 R   ps
```

In this case the fee-server is running with Process Ids (PID) 972, 973, 974, while the script that started the fee-server has PID 971. Please note that the PIDs will vary.

J.4.8.2 If the fee-server was started automatically from the gateway

If the DCS cards are on the private network on the DCS gateway then the fee-servers should start up automatically after rebooting the DCS cards, or powering up the RCU and the list of processes will look slightly different.

```
alphsdcs0282:/ \$ ps -a
 PID  Uid     VmSize Stat Command
   1 root        356 S   init
   2 root            SW  [keventd]
   3 root            SWN [ksoftirqd_CPU0]
   4 root            SW  [kswapd]
   5 root            SW  [bdflush]
   6 root            SW  [kupdated]
   7 root            SW  [mtdblockd]
   8 root            SWN [jffs2_gcd_mtd3]
  35 root        340 S   /sbin/syslogd -m 0 -O /var/log/messages
  36 root        296 S   /sbin/klogd
  38 root        396 S   /sbin/devfsd /dev
  69 root        360 S   udhcpc -n
  75 root        908 S   /usr/local/sbin/sshd -D
  80 root        352 S   /usr/sbin/inetd
  81 root        360 S   /usr/local/sbin/boa
  85 root        348 S   /usr/sbin/crond -l 9 -c /etc/crontab/
  91 root        216 S   /usr/local/sbin/writer
 116 root            SW  [rpciod]
 137 root        384 S   /bin/sh /mnt/master/fee-net/bin/startFeeServer.sh alphsdcs0282
 138 root        416 S   -sh
 158 root        392 S   /mnt/master/fee-net/bin/feeserver
 159 root        392 S   /mnt/master/fee-net/bin/feeserver
 160 root        392 S   /mnt/master/fee-net/bin/feeserver
 359 root       1160 S   /usr/local/sbin/sshd: root@pts/0
 361 root        416 S   -sh
 365 root        340 R   ps -a
```

The fee-server will be mounted from a network disk (on the gateway) and will in general not be the same as the one that it started manually from the DCS card. In the list above there are three processes associated with the fee-server on is the start-up script */bin/sh /mnt/master/fee-net/bin/startFeeServer.sh alphsdcs0282* with PID 137, and three processes associated with the fee-server itself. In this example they have PIDs 158, 159 and 160. These start-up script monitors the fee-server an automatically restarts it if is stopped. Therefore if one wants to kill the fee-server it is necessary to first kill the start-up

script.

J.4.9 Start a fee-server on the DCS card

To start the DIM servers first log into the card where you want to start the fee-server (Section J.4.7). To start the fee-server do the following:

```
[any machine ~] > ssh root@alphsdcsXXXX ssh

root@alphsdcsXXXX's password: dcs

alphsdcsXXXX:\$ ./home/phos/feeserver_start.sh
```

Here XXXX represents the serial number of the DCS card. If for instance you are logged in to *alphsdcs0279* and wants to start a fee-server there, then do the following:

```
[any machine ~] > ssh root@alphsdcs0279 ssh

root@alphsdcs0279's password: dcs

alphsdcs0279:\$ ./home/phos/feeserver_start.sh
```

When the fee-server is started it will issue several lines relevant (and irrelevant) information on the command line. If the fee-server was started up successfully the last few lines of information should look exactly like this.

```
-----------------------------------
init services for configuration PHOS
... done
Init OK
DIM Server successfully started, ready to accept commands.
```

Anything else means that there i some kind of problem. To solve them go to Chapter J.6.3.

J.4.10 Kill a fee-server running on the DCS card

J.4.11 Kill a fee-server that was started manually

If the fee-server was started manually it is two ways to kill it. In the following discussion XXXX Will refer to the serial number of the DCS card. Before you kill a fee-server please talk to your colleagues so you don't interfere with their work.

- From the terminal window where the fee-server was started: press **CTRL + C** on the PC key board. No you should get back the normal command prompt an the last few line should look either like something like this.

```
CE Info: (../src_ce/ce_base.c line 592) program rm exited with error code 0
CE Debug: issue result buffer: 0x30377830
Retcode of CMND timedwait: 0
Issue-nRet: 0
ErrorCode in Header: 0
ACK

alphsdcsXXXX:/ \$
```

  Or something like this

```
-------------------------------------
init services for configuration PHOS
... done
Init OK
DIM Server successfully started, ready to accept commands.

alphsdcsXXXX:/ \$
```

- If you want to kill the fee-server from another terminal window than where the fee-server was started do the following. First log in to the DCS card where the fee-server you want to kill is running (see Section J.4.7). To kill the fee-server process you need to know the Process Id (PID). See Section J.4.8 for how to obtain the PID. If a fee-server is running there will be three processes called **/bin/feeserver** To kill all processes associated with the fee-server it will be sufficient to kill only the first one. To kill it do the following on the command line.

```
alphsdcsXXX:/ \$ kill -9 XXX
```

  Where **XXX** represents the PID of the process. For the example given in Section J.4.8, XXX = 972 an you would kill it with the command

```
alphsdcsXXX:/ \$ kill -9 972
```

In addition There is currently two more ways to kill the fee-servers. One is to turn on and off the power of the DCS/RCU card. The other one is to reboot the DCS card. Please not that in the near future the fee-server will start up automatically on reboot or power up an it will not be possible to kill the fee-server by turning on/off power or by rebooting. If you want to kill the fee-server by rebooting do the following:

```
alphsdcsXXXX:/ \$ reboot
```

After this command you will loose contact with the DCS card. The DCS card will use ∼1 min to reboot. Shortly after the **reboot command** the command prompt should look like this.

```
alphsdcsXXXX:/ \$ Connection to alphsdcsXXXX closed by remote host.
Connection to alphsdcsXXXX closed.
[any PC ~] >
```

J.4.12 Kill a fee-server is running on the Gateway (started automatically)

If the fee-server was started automatically at boot up of the DCS card, then in order to kill it you must first kill the process that are monitoring the fee-server, please refer to Section J.4.7. If you try to kill the fee-server without first killing the monitoring process then the monitoring process will immediately detect that the fee-server is not running and restart it. After you have killed the script that started the fee-server, then kill the fee-server as described in the previous section.

J.5 Log in to the PC where the PABC and the DIM DNS server is running

The PABC GUI, the DIM DNS server and DID will all run on the PC *alphspcdcs01.cern.ch*. This PC is currently situated in the corner behind the PHOS module in 167 R-006. You can either log in directly to this PC with user name **phos** and password **phosbeamtest** or remotely from another machine. To log in remotely via the network do the following:

```
[any PC ~] >   ssh phos@alphspcdcs01
Scientific Linux CERN Release 3.0.6 (SL)
phos@alphspcdcs01's password: phosbeamtest
[alphspcdcs01] /home/phos >
```

J.5.1 Check if the DIM DNS server is running

On the command prompt of the PC *alphspcdcs01* type the command **ps -e** [4] You will get long list of processes that looks something like the list below (not all the processes)

```
[alphspcdcs01] /home/phos > ps -e
PID TTY          TIME CMD
1 ?        00:00:05 init
2 ?        00:00:00 migration/0
3 ?        00:00:00 migration/1
      ....... more processes
2982 ?        00:00:00 rpciod
2986 ?        00:00:00 rpc.mountd
2995 ?        00:00:00 dhcpd
3015 ?        00:00:05 sendmail
3024 ?        00:00:00 sendmail
3034 ?        00:00:04 gpm
3043 ?        00:00:00 crond
3066 ?        00:00:00 xfs
3083 ?        00:00:00 atd
3111 ?        00:00:28 dimNameServer
3114 tty1     00:00:00 mingetty
3115 tty2     00:00:00 mingetty
        ...... more processes
```

If the DIM DNS server is running there should be either a process with the name **dimNameServer** or with the name **dns**. If the DNS server was stared (as it should) automatically the name will be **dimNameServer** and if it was started manually from the command line it will have the name **dns**. In this examples the DNS server is running with PID 3111.

J.5.2 Start the DIM DNS server

To start the DIM DNS server you need to first log in to the PC *alphspcdcsp1*. You must run a script to set up the environment. From the command prompt do the following:

```
[alphspcdcs01] /home/phos > . dcs_cern.sh

[alphspcdcs01] /home/phos > dns
PID 11250 - Sat May 27 18:08:06 2006 -  DNS (pid 11250) starting up on alphspcdcs01.ce
```

[4] the **e** stands for *every* process running on the PC as opposed to the option **-a** that means *all*processes except session leaders and processes not associated with a terminal.

The response from the DIM DNS server should be exactly as sown above except that the date and the PID will be different. If you get any other message go to section J.6.4 *alphspcdcs01* (see Section J.5)

J.5.3 Kill the DIM DNS server

The only reason to kill the DNS server is for debugging purposes when the DCS system is not working properly and you want to see the messages from the DNS server on the command line. In this case you kill it and then restart it manually from the command line.

There is three ways to kill the DIM DNS server. Before you kill the DIM DNS server please talk to your colleagues so you don't interfere with their work.

- To kill the DIM DNS server from the terminal window where you started the DIM DNS server type **CTRL +C** in the terminal window where the DIM DNS server is running.
- If you are not in the terminal window where the DIM DNS was stared you will first have to log in to the PC running the DIM DNS server. (see Section J.5). To obtain the PIDs of the processes do the following:

  ```
  [alphspcdcs01] /home/phos > ps -e
  ```

  You will get list of processes similar to that given in section J.5.1. Find the PID and kill it with the command

  ```
  /home/phos > kill -9 XXXX
  ```

  where XXX is the PID. If for instance the PID is 3111 as in section kill it by typing

  ```
  /home/phos > kill -9 3111
  ```

J.5.4 Kill the PABC GUI

If you are in the terminal window where you started the PABC GUI just press **CTRL + C** on the PC keyboard. Otherwise kill the PABC GUI by logging in remotely from another PC to the machine that is running the PABC GUI and do the following:

```
[any PC ~] >  ps -a
```

The list of processes should look something like this

```
[any PC ~] > ps -a
    PID TTY          TIME CMD
  25005 pts/7    00:00:01 apdgui
   8456 pts/14   00:00:00 ps
```

The executable of the PABC has the name “apdgui”. In this examples it has the PID 25005. To kill it do the following.

```
[any PC ~] > kill -9 25005
```

J.5.5 Kill a process i general

To kill a process in general type the command **ps -a** on the command prompt in the PC/DCS card where the process you want to kill is running. Look at the PID of the process you want to kill ant type on the command line

```
[any PC ~] > kill -9 PID
```

J.6 Troubleshooting

J.6.1 I cannot take any data

Check if triggers are arriving at the RCUs First of all check if there are triggers arriving to the RCU and also read the error register (0x7800) the trigger configuration register (0x7801) and the trigger counter register (0x78002). You can read all three register with a single command.

```
alphsdcs0279:/  rcu-sh r 0x7800 3
0x7800: 0 0x1410160 0x1780178
```

Repeat the procedure for the RCUs you have activated for data taking. The result should look something like the one above. Here there are no errors (0x7800 = 0x0). The trigger configuration is hardware trigger and time window to L2 trigger is 0x160 clock cycles. The last register says that 0x178 L1 triggers has arrived to the RCU (most significant 16 bits) and that 0x178 L1 triggers was accepted least significant 16 bits. If you read out the trigger counter register several times and there are trigger then the counter will increase its value. If triggers arrives in the RCU and you cannot take any data then the problems is somewhere else

Check if the RCU is in error Read the status and error register of the RCU

```
alphsdcs0279:/  rcu-sh r 0x7800
0x7800: 0x80000000
```

The value of this register should be zero. If you get anything else the RCU must be reset. This can be done by clicking the **Disarm Trigger** button on the PABC GUI. There are several conditions that can leave the RCU in some state of error. Some of the are listed below.

- The trigger rate is to high for the RCU

  Measure increase the trigger blocking (dead) time.

- Date was stopped before disarming triggers.

  Measure click the **Disarm Trigger** button and repeat the data taking procedure.

- Triggers was enabled before DATE was started,

  Measure click the **Disarm Trigger** button and repeat the data taking procedure.

Please note that the trigger blocking time necessary depends on how many channels you want to read out. To read out the whole module the trigger blocking time should be at least 5 ms. To read out a 7x7 matrix the blocking time should be around 200 - 300 us.

Check that you have enables the correct LDCs in DATE Make sure that only the DATE Equipments that you want data from is activated. If you activate an equipment in DATE but the readout region you want does not include any readout channels from that Equipment (RCU) then date will still expect data from it and will wait forever for data from that equipment. There are four DATE equipments per PHOS Module. The DATE name of these equipments is DDL8, DDL9, DDL10, DDL11 and they corresponds to RCU0 (alphsdcs0060), RCU1 (alphsdcs0279) RCU2 (alphsdcs0281) and RCU3 (alphsdcs0282) respectively. To activate/deactivate equipments issue the command **editDb** on the command line of the DAQ computer (currently *aldaqpc019*).

J.6.2 I cannot turn on the front end cards

Checklist.

- Make sure that the front end card you want to turn ON is actually connected.
- Check if the fee-server is running J.4.8

J.6.2.1 Interpretation of messages from the LogViewer

J.6.2.2 Messages when applying APD bias settings

See Section J.4.2 and J.6.7.

J.6.2.3 Messages when Arming trigger

J.6.3 Error when starting a fee-server (DIM server)

Interpretation of error messages:

- ```
  DIM: [ERROR] - Connecting to DIM_DNS on alphspcdcs01.cern.ch: Connection refuse
  DIM Server successfully started, ready to accept commands.
  ```

  This message from the fee-server means that fee-server started correctly, but did not manage to contact the DIM DNS server.
  Possible reasons:

  The DIM DNS server is not running: See Chapter J.5.1 for how to check if the DIM DNS server is running, and Chapter J.5.2 for how to start it if it is not running.

  The network cable is not connected properly
  Go to the PC *alphspcdcs01* and check if the Ethernet cable is properly connected bot tho the PC and to the switch.

- ```
  DIM Server successfully started, ready to accept commands.
  DIM: [FATAL] - alphsdcs0281: Some Services already known to DNS.
  Framework tries to exit FeeServer (3)
  Most likely FeeServer name already exists.
  Exit state: 204

  CE Info: (../src_ce/rcu_service.c line 104) releaseRcuAccess finished with erro
  ```

  This message means that a fee-server with the same name as the one you try to start is already running. Either you have tried to start a fee-server twice on the same DCS card, or a fee-server with the same name is already running on another DCS card (Se Chapter J.4.8).

J.6.4 Error when starting DIM DNS server

Interpretation of error messages:

```
PID 11372 - Sat May 27 18:12:09 2006 -  DNS (pid 11372) starting up on alphspcd
PID 11372 - Sat May 27 18:12:09 2006 - (ERROR) Opening server port: Address alr
```

This message means that the DIM DNS is already running. The DIM DNS server uses port 2025 and port 5100. The DIM DNS server starts and then kill itself because it finds that these ports are already in use.
Measure: Most likely everything is OK, it is just not possible to run two DIM DNS servers on the same PC. If you really wants to make sure that everything is OK then see Chapter J.5.3 and Chapter J.5.2 for how to kill the server that is running and to start your own. Before you kill the DIM DNS server talk to your colleagues so that you don't interfere with their work.

```
[alphspcdcs01] /home/phos > dns
-bash: dns: command not found
```

Reason: you have forgotten to set up the environment variable. To set up the environment do.

```
[alphspcdcs01] /home/phos > . dcs_cern.sh
```

Don't forget the dot + space.

J.6.5 Error message when starting DID

```
[alphspcdcs01] /home/phos > did
PID 4070 - Sun May 28 13:18:51 2006 -
(ERROR) Connecting to DIM_DNS on alphspcdcs01.cern.ch: Connection refused
```

This message means that you have forgotten to start DIM DNS server. See Chapter J.5.2 for how to start the DIM DNS server.

DID crashes It can happen that DID crashes if you run it remotely due to a bug in DID. You will get an error message about *segmentation fault*. If this happen you can try to run DID directly on the PC.

J.6.6 Error when starting the PABC GUI

When starting the PABC GUI the GUI looks strange for several minutes and then it appear correctly:
Reason: The PABC GUI is not able to contact the DIM DNS server. The timeout period is several minutes and in the meantime the PABC will hang. Either the DIM DNS server was not started or there are some network problems. If you are running the PABC GUI (apdgui) on the PC *alphspcdcs01.cern.ch* then the reason is that the DIM DNS server was not started. See Chapter J.5.2 for how to start the DIM DNS server.

```
ERROR could not turn on CardX at branchY, Reason: DCS is not master.
```

There is several possible reasons for this message.

- The RCU was not reset after it was powered up.
  measure: Reset the RCUs. See Chapter J.4.6 for details
- The fee-server is not running.
  measure: Check if the fee-server is running (Chapter J.4.8). If it is not running the start the fee-server (Chapter J.4.9).
- Somebody have changed the access rights of DCS by for instance running scripts from the command line of the DCS card.
  measure: Reset the RCUs (Chapter J.4.6).

If none of the measure does not solve the problem it might be necessary to set up the system from scratch. See Chapter J.2 for details.

J.6.7 Error when applying APD settings

J.6.7.1 ERROR 1: No response from FEE

When attempts is made to apply APD settings to the FEE the LogViewer issues the message

```
applying APD values for cardX at branchY ...ERROR 1: No response from FEE
```

This message means that the PABC is not able to establish contact with the given FEEC.
possible reasons:

- FEEC is not turned ON.
  measure: Turn ON the FEEC (Section J.4.1).
- The FEEC has no power.
  measure: Turn on power for the FEEC.

- The FEEC is not present.
  measure: Connect a FEEC on the corresponding slot on the GTL back-plane and turn it ON (Section J.4.1).

J.6.7.2 ERROR 3: DCS is not master

When attempts is made to apply APD settings to the FEE the LogViewer issues the message.

```
applying APD values for cardX at branchY ERROR 2: DCS is not master
```

The message means that DCS is not master of the RCU bus an is therefore not able to set the APD DAC values.
Reason:
Somebody has taken the control away from DCS by running scripts on the command line of the DCS card.
Measure: Reset the RCU (Chapter J.4.6).

J.6.7.3 ERROR 3: Read-back values are crazy

When attempts is made to apply APD settings to the FEE the LogViewer issues the message

```
applying APD values for cardX at branchY...ERROR 3: Readback values are crazy
```

This means that the APD values the PABC GUI reads back from the electronics does not make sense at all.
Possible reasons:

- The fee-server is not running (see Chapter J.4.8.
  Measure: Reset the RCUs (see Chapter J.4.6)

- The RCU is in some state of fatal error because somebody ran scripts on the command line.
  Measure: Reset the RCUs (see Chapter J.4.6).

In none of these measure solves the problem it might be necessary to set up the communication chain from scratch (Chapter J.2).

J.6.8 I cannot log in to the DCS card

J.6.9 No route to host..

When trying to log in to the DCS card you get an error message that looks something like this

```
ssh: connect to host alphsdcs0279 port 22: No route to host
```

One reason might be that you are trying to log in to the DCS cards from another machine than the DCS gateway *alphspcdcs01*, while the DCS cards are running at a private subnetwork on the gateway. In this case the DCS cards has received the IP address from a dedicated DHCP server on the gateway instead of from the CERN DHCP server. Therefore, when you try to log in to the DCS card from the CERN network it is not possible to route the request to the DCS cards (no route to host) because the CERN name server does not know the IP address of the DCS cards.
Measure: First log in to the gateway, and then log in to the DCS cards from the gateway. See section J.4.7 for details.

Another reason might be that there were some problems with the Ethernet connection during boot up of the DCS card. The DCS card request an IP address from the DHCP server at boot time. Therefore, if there are problems with the Ethernet connection at boot time then it might be that the DCS card has not received a valid IP address.
Measure: Reboot the DCS card by turning ON and OFF the power of the RCU.

J.6.10 Connection timed out..

When trying to log in to the DCS card you get an error message that looks something like this

```
ssh: connect to host alphsdcs0279 port 22: Connection timed out
```

This means that the request for login is routed correctly to the DCS card, but that the DCS card is not responding.
Checklist

- Check that the power for the RCU is ON.
- Check that the Ethernet cable is properly connected to the DCS card and the switch.
- Check that the switch is connected properly to the bottom network card[5] of *alphspcdcs01*

[5]This holds if the DCS cards are running on the Gateway, if they run openly on the CERN network, then check that the switch is connected properly to a fan-out

After performing these checks then **reboot** the DCS cards by turning ON and OFF power. If the DCS cards has Ethernet connection there will be red light on the DCS board, there should also be another red light blinking. The blinking indicates that the DCS cards receives TCP/IP packages.

J.6.10.1 Permission denied....

When trying to log in to the DCS card you get an error message that looks something like this

```
Permission denied, please try again.
```

This message means that the password does not match the user.
Measure: Check that you are logging in as user **root** with password **dcs**.

J.7 Conversion between DAC and High Voltage

Table J.7 shows the correspondence between the DAC values and High Voltage. Please note that these values are only rough estimates. Actual HV on the APD might vary with several Volt relative to the values given in the Table.

Table J.1: Correspondence between DAC values and High Voltage

DAC value	Voltage/Volt
0	210
30	215
60	221
90	226
120	232
150	237
180	243
210	249
240	254
270	260
300	265
330	271
360	276
390	282
420	288
450	293
480	299
510	304
540	310
570	315
600	321
630	327
660	332
690	338
720	343
750	349
780	354
810	360
840	366
870	371
900	377
930	382
960	388
990	393
1020	399
1023	400

J.8 Software Documentation

The the PHOS DCS software and HTML documentation is available for download at

`http://folk.uio.no/perthi/alice/doc/phos_dcs.tar`

The HTML documentation for the source code can also be viewed on-line at

`http://folk.uio.no/perthi/alice/doc/phos_dcs/html/`

Bibliography

[1] Source: Compressed Baryonic Matter (CBM) Experiment at the Facility for Antiproton and Ion Research (FAIR), GSI, Darmstadt, Germany.

[2] Anders Strand Vestbø. *Pattern Recognition and Data Compression for the ALICE High Level Trigger*. PhD thesis, University of Bergen, Norway, May 2004.

[3] K. Reygers (PHENIX Collaboration). *J.Phys.* **G 35** *104045*, 2008.

[4] J. Adams et al. (STAR collaboration). *Phys.Rev.Lett.* **91**, *072304*, 2003.

[5] A. Adare et al. (PHENIX Collaboration). *arXiv:0804.4168v1 [nucl-ex]*, 2008.

[6] Cern accelerator complex `http://documents.cern.ch/cgi-bin/setlink?base=PHO&categ=photo-di&id=0606052`, C. Vanoli, 2006.

[7] Christian Holm Christensen. *ALICE Forward Multiplicity Detector, From Design to Installation*. PhD thesis, Niels Bohr Institute, University of Copenhagen, December 2007.

[8] ALICE Collaboration. CERN/LHCC-2003-062 ALICE Technical Design Report, Trigger, Data Acquisition, High Level Trigger, Control System. Technical report, 2003.

[9] Matthias Richter et. al. High Level Trigger Applications for the ALICE Experiment. *IEEE Transactions on, Nuclear Science*, 55(1), February 2008.

[10] Johan Alme. *Firmware Development and Integration for ALICE TPC and PHOS Front-end Electronics*. PhD thesis, University of Bergen, Norway, July 2008.

[11] *PHOS basics for the user*. PHOS internal document avilable at `http://folk.uio.no/perthi/alice/doc/PHOS-User-Manual.pdf`.

[12] ALICE PHOS collaboration. A high resolution electromagnetic calorimeter based on lead-tungstate crystals. *Nuclear Inst. and Methods in Physics Research, A*, 550(1), September 2005.

[13] A. Badalà, F. Blanco, P. La Rocca, F. Librizzi, G. S. Pappalardo, A. Pulvirenti, F. Riggi, R. Vernet, T. C. Awes, H. Muller, and N. Tupikin. Characterization of avalanche photodiodes (APDs) for the electromagnetic calorimeter in the ALICE experiment. *Nuclear Instruments and Methods in Physics Research A*, 596:122–125, October 2008.

[14] ALICE Collaboration. ALICE Technical Design Report of the Photon Spectrometer (PHOS). Technical report, 1999.

[15] ALICE Collaboration. ALICE: Physics Performance Report, Volume 1. *Journal of Physics G, Nuclear and Particle Physics*, 19(11), September 2004.

[16] Christoph Scholz Bogdan Povh, Klaus Rith and Frank Zetsche. *Particles and Nuclei*. Springer, 2003. ISBN 3-540-20168-8.

[17] J. D. Bjorken. *Phys. Rev. D27*, 35:140–151, 1983.

[18] Cheuk-Yin Wong. *Introduction to High-Energy Heavy-Ion Collisions*. Wordl Scientific, 2007. ISBN 981-0-202-636.

[19] Raimond Snellings Sergei A. Voloshin, Arthur M. Poskanzer. Collective phenomena in non-central nuclear collisions. 2008. arXiv:0809.2949v2 [nucl-ex].

[20] G.E. Bruno (NA57 Collaboration) and references therein. *J.Phys.* **G 30**, *717, 072304*, 2004.

[21] M.C. Abreu et al. (NA50 collaboration). *Phys. Lett.* **B477**, 2000.

[22] A. Adare et al. (PHENIX Collaboration). *Phys.Rev.Lett.* **98**, *232301*, 2007.

[23] T. Peitzmann and M.H. Thoma. *Phys. Rept.* **364**, *175*, 2002.

[24] T. Sakaguchi and references therein. *J. Phys.* **G 35**, *104025*, 2008.

[25] D.K. Srivastava and references therein. *J. Phys.* **G 35**, *104025*, 2008.

[26] C. Gale and K.L. Haglin D.K. 2003. Invited review for QGP3, edited by R. Hwa and X.-N. Wang, hep-ph/0306098.

[27] R.J. Fries et al. *nucl-th/0707.0261*, 2007.

[28] ALICE Collaboration. ALICE: Physics Performance Report, Volume 2. *Journal of Physics G, Nuclear and Particle Physics*, 32(11), November 2006.

[29] ALICE Collaboration. ALICE Electromagnetic Calorimeter Technical Design Report. Technical report, 2008.

[30] ALICE Collaboration. ALICE Technical Design Report of the Time Projection Chamber (TPC). Technical report, 2000.

[31] ALICE Collaboration. ALICE Technical Design Report of the Transistion Radiaton Detector (TRD). Technical report, 2001.

[32] ALICE Collaboration. ALICE Technical Design Report of the Time of Flight System (TOF). Technical report, 2000.

[33] ALICE Collaboration. ALICE Technical Design Report of the Photon Multiplicity Detector (PMD). Technical report, 1999.

[34] ALICE Collaboration. ALICE Technical Design Report of the Dimuon Forward Spectrometer. Technical report, 1999.

[35] ALICE Collaboration. ALICE Technical Design Report of the Inner Tracking Systemm (ITS). Technical report, 1999.

[36] ALICE Collaboration. ALICE Technical Design Report of the Zero Degree Calorimeter (Z0). Technical report, 1999.

[37] ALICE Collaboration. ALICE Technical Design Report of the Detector for High Momentum Particles (HMPID). Technical report, 1998.

[38] ALICE Collaboration. ALICE Technical Design Report on Forward Detectors: FMD, T0 and V0. Technical report, 1998.

[39] ALICE Collaboration. Addendum to the ALICE Technical Design Report of the Time of Flight System (TOF). Technical report, 2002.

[40] ALICE Collaboration. Addendum to the ALICE Technical Design Report of the Dimuon Forward Spectrometer. Technical report, 2000.

[41] M.Richter T.Steinbecj J.Thaeder J.Wagner, V.Lindenstruth. Lossless Data Compression for ALICE HLT. Technical report, 2008.

[42] B.R. Martin and G. Shaw. *Particle Physics*. John Wiley & Sons, 2003. ISBN 0-471-97285-1.

[43] P.T. Hille. Time of Flight resolution of a prototype ALICE Photon Spectrometer (PHOS) module. Master's thesis, Department of Physics, University of Oslo, January 2005.

[44] *ALICE TPC Readout Chip User Manual*. CERN - EP/ED, June 2002.

[45] *ALICE TPC Readout Chip User Manual*. CERN - EP/ED, June 2002.

[46] W. R. Leo. *Techniques for Nuclear and Particle Physics Experiments.* Springer, 1994. ISBN 0-387-57280-5.

[47] Matthias Richter. *Development and Integration of on-line Data Analysis for the ALICE Experiment (Unfinnished).* PhD thesis, University of Bergen, Norway, August 2009.

[48] *RCU Firmware: Registers and Commands.* CERN - EP/ED, June 2002.

[49] Vasilie Buzuloiu. Real time recovery of the amplitude and shift of a pulse from its samples. *CERN/LAA/RT92-015*, April 1992.

[50] Vasilie Buzuloiu et al. Optimal recovery of signal parameters from a few samples: one and two-dimensional applications. *Optical Engineering*, 35(6):1576–1582, June 1996.

[51] F. Chuman et.al. Calibration Methods of PHOS Modules. Technical report, 2008.

[52] Zhou Daicui Yin Zhongbao Wan Renzhuo, Xu Chuncheng. Track Matching Styudy from TPC to PHOS in ALICE. Technical report, 2007.

[53] *The ALICE Offline Bible.* CERN - EP/ED, Version 0.00 2007.

[54] Private communication with Alexei Kuryakin.

[55] Particle Data Group. Review of particle physics. 2008.

[56] Private communication with Johan Alme.

[57] Private communication with Torsten Alt and Johan Alme.

[58] Private communication with Christian Lippman.

[59] Øystein Djuvsland. PHOS Commissioning and Noise Status. Internal ALICE presentation, December 2008.

[60] HAMAMATSU. Characteristics and use of Si APD (Avalanche Photodiode). Technical report, May 2004.

[61] Monson H Hayes. *Statistical Digital Signal Pocessing and Modeling.* John Wiley & Sons, 1996. ISBN 0-741 59431-8.

[62] Gouri K Bhattacharyya and Richard A Johnson. *Statistical Concepts and Methods.* John Wiley & Sons, 1977. ISBN 0-471-07204-4.

[63] Fredric James. *Statistical Methods in Experimental Physics, Second Edition.* Duxbury Press, 2006. ISBN-13 978-981-256-795-6.

[64] D. Marquardt. An algorithm for lest-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[65] R.K Bock H. Grote D.Notz R. Fruhwirth, M. Regeler. *Data Analysis Techniques for High-Energy Physics, Second Edition*. Camebridge, 2000. ISBN-0-521-63584-9.

[66] Private communication with David Gesbert, Univeristy of Oslo.

[67] http://dim.web.cern.ch/dim/. The DIM web-site.